



Politechnika Gdańska
Wydział Elektroniki, Telekomunikacji
i Informatyki
Katedra Inżynierii Oprogramowania



Teresa Zawadzka

Integracja heterogenicznych źródeł wiedzy z wykorzystaniem logiki opisowej

Rozprawa doktorska

Promotor:

dr hab. inż. Krzysztof Goczyła,
prof. nadzw. Politechniki Gdańskiej

Gdańsk, 2008

Praca współfinansowana przez Ministerstwo Nauki i Szkolnictwa Wyższego
w ramach grantu promotorskiego nr N516 036 31/3501.

Podziękowania

Chciałabym serdecznie podziękować Rodzicom, przede wszystkim za ich wsparcie na całej mojej drodze edukacji, od trudnych początków w szkole podstawowej aż do dzisiaj, kiedy to składam moją pracę doktorską. To również Oni rozbudowali we mnie zamiłowanie do nauk ścisłych, a w sposób szczególny do matematyki, za co jestem Im ogromnie wdzięczna.

Dziękuję również mojemu Mężowi, który był również jednym z moich najbliższych współpracowników w projekcie PIPS, który z ogromną cierpliwością wysłuchiwał moich pomysłów oraz wyręczał mnie w ich implementacji.

Dziękuję Wojtkowi Waloszkowi, który zawsze dawał inspirację do nowych pomysłów i który jest twórcą kartograficznej reprezentacji wiedzy, bez której metody zaproponowane w rozprawie nigdy by nie powstały.

W sposób szczególny jednak dziękuję Profesorowi Krzysztofowi Goczyle za opiekę promotorską, za Jego pracę włożoną w obecny kształt niniejszej rozprawy, ale przede wszystkim za umiejętność stawiania wymagań, oceniania pracy i wsparcia w (niestety nieuniknionych) niepowodzeniach.

Spis treści

PODZIĘKOWANIA	III
SPIS TREŚCI.....	IV
WYKAZ SYMBOLI I OZNACZEŃ.....	VI
ROZDZIAŁ 1. WSTĘP	1
1.1 JAK DANE STAJĄ SIĘ WIEDZĄ.....	1
1.2 OD SIECI WWW DO SIECI SEMANTYCZNEJ.....	2
1.3 CELE I TEZA PRACY	3
1.4 UKŁAD ROZPRAWY	3
1.5 KONWENCJE NOTACYJNE	5
ROZDZIAŁ 2. WPROWADZENIE W ZAGADNIENIA INTEGRACJI WIEDZY W SIECI SEMANTYCZNEJ	6
2.1 LOGIKA OPISOWA JAKO FORMALNY SPOSÓB ZAPISU ONTOLOGII	6
2.2 ONTOLOGIE A LOGIKA OPISOWA.....	9
2.3 WNIOSKOWANIE I SYSTEMY REPREZENTACJI WIEDZY	9
2.4 INTEGRACJA W SIECI SEMANTYCZNEJ	10
2.5 SYSTEM INTEGRACJI WIEDZY W SIECI SEMANTYCZNEJ	17
2.6 PODSUMOWANIE	24
ROZDZIAŁ 3. SYSTEM INTEGRACJI WIEDZY	25
3.1 MODEL SYSTEMU INTEGRACJI WIEDZY	25
3.2 JĘZYK ZAPYTAŃ DO SYSTEMU INTEGRACJI WIEDZY	34
3.3 PODSUMOWANIE	37
ROZDZIAŁ 4. KARTOGRAFICZNA REPREZENTACJA WIEDZY PODSTAWĄ ALGORYTMÓW SYSTEMU INTEGRUJĄCEGO WIEDZĘ	38
4.1 KARTOGRAFICZNA METODA REPREZENTACJI WIEDZY	38
4.2 PODSUMOWANIE	39
4.3 ŹRÓDŁO WIEDZY <i>KNOWLEDGE SIGNATURE ANALYSER – KASa</i>	41
4.4 PODSUMOWANIE	42
ROZDZIAŁ 5. METODA TWORZENIA ŹRÓDEŁ WIEDZY PRZEZ SEMANTYCZNE WZBOGACANIE ŹRÓDEŁ DANYCH.....	43
5.1 ONTOLOGICZNY OPIS DANYCH	43
5.2 BUDOWANIE ŹRÓDŁA WIEDZY DLA DANYCH OPISANYCH ONTOLOGICZNIE	51
5.3 PODSUMOWANIE	57
ROZDZIAŁ 6. METODA PRZETWARZANIA ZAPYTAŃ W ŚRODOWISKU ROZPROSZONEGO SYSTEMU INTEGRACJI WIEDZY	58
6.1 PRZETWARZANIE ZAPYTAŃ W ŚRODOWISKU ROZPROSZONYM.....	58
6.2 ZALEŻNOŚCI POMIĘDZY TERMINOLOGIAM W SYSTEMIE INTEGRACJI WIEDZY	58
6.3 DOSTOSOWANIE ALGORYTMÓW WNIOŚKUJĄCYCH Z WIEDZY REPREZENTOWANEJ KARTOGRAFICZNIE DO WARUNKÓW JEJ ROZPROSZENIA	62
6.4 PODSUMOWANIE	67
ROZDZIAŁ 7. ZASTOSOWANIE METOD INTEGRACJI WIEDZY W SYSTEMIE KISS – KNOWLEDGE INTEGRATION SUPPORT SYSTEM	69

7.1	OGÓLNA ARCHITEKTURA SYSTEMU KISS (<i>KNOWLEDGE INTEGRATION SUPPORT SYSTEM</i>)	69
7.2	KASEA (<i>KNOWLEDGE SIGNATURE ANALYSER</i>).....	70
7.3	KL (<i>KNOWLEDGE LAYER</i>)	71
7.4	EQM (<i>EXTERNAL QUERY MANAGER</i>).....	74
7.5	PODSUMOWANIE	75
ROZDZIAŁ 8. OCENA PRZEDSTAWIONYCH METOD INTEGRACJI WIEDZY		76
8.1	SYSTEM PIPS (<i>PERSONALISED INFORMATION PLATFORM FOR LIFE AND HEALTH SERVICES</i>)	76
8.2	USŁUGA DOBORU I WSPOMAGANIA STOSOWANIA OPTYMALNEJ DIETY.....	79
8.3	ANALIZA ZŁOŻONOŚCI OBLICZENIOWEJ ALGORYTMÓW	87
8.4	PODSUMOWANIE	88
ROZDZIAŁ 9. KIERUNKI DAJSZYCH PRAC		89
9.1	ROZWÓJ MODELU INTEGRACJI WIEDZY	89
9.2	METODY BUDOWANIA ŹRÓDEŁ WIEDZY PRZEZ WZBOGACENIE WARSTWY DANYCH O WARSTWĘ WIEDZY.....	98
9.3	ALGORYTMY PRZETWARZANIA ZAPYTAŃ W SYSTEMIE INTEGRACJI WIEDZY	110
9.4	PODSUMOWANIE	110
ROZDZIAŁ 10. PODSUMOWANIE		111
SŁOWNIK POJĘĆ		116
SPIS DEFINICJI		123
BIBLIOGRAFIA		126

Wykaz symboli i oznaczeń

\mathcal{A}_O	alfabet ontologii
\mathcal{A}_{O_A}	alfabet nazw atrybutów w ontologii
\mathcal{A}_{O_C}	alfabet nazw konceptów w ontologii
\mathcal{A}_{O_I}	alfabet nazw osobników w ontologii
\mathcal{A}_{O_R}	alfabet nazw ról w ontologii
C, D	koncepty
C^I	interpretacja konceptu
$C \sqcap D$	przecięcie konceptów
$C \sqcup D$	suma konceptów
$C \sqsubseteq D$	zawieranie (subsumcja) konceptów
$C \equiv D$	równość konceptów
$C(a)$	asercja wystąpienia konceptu
\mathcal{D}	dziedzina konkretna
\mathcal{G}	ontologia globalna
\mathcal{H}_T	hierarchia terminologii
\mathcal{I}	interpretacja
\mathcal{K}	baza wiedzy
\mathcal{KS}	źródło wiedzy
\mathcal{L}_O	logika opisowa
$\mathcal{M}_{\mathcal{G}, S}$	odwzorowania
\mathcal{M}_A	odwzorowanie atrybutu
\mathcal{M}_C	odwzorowanie konceptu
\mathcal{M}_R	odwzorowanie roli
$\mathcal{M}_{s(C)}$	odwzorowanie sygnatury konceptu
\mathcal{O}	ontologia
$\mathcal{O}[O_x, S]$	ontologia dla zbioru odwzorowań S
\mathcal{O}_x	terminologia
\mathcal{O}_s	opis świata
$\mathcal{O}_s[S]$	zbiór asercji dla zbioru odwzorowań S
\mathcal{Q}_O	język zapytań
\mathcal{Q}_{O_T}	zapytania terminologiczne
\mathcal{Q}_{O_S}	zapytania dotyczące osobników
$pred(\mathcal{D})$	nazwy predykatów dla dziedziny \mathcal{D}
q	zapytanie
r	odpowiedź na zapytanie
R, S	role

R^I	interpretacja roli
R^-	inwersja roli
R^+	domknięcie przechodnie roli
$R \equiv S$	równość ról
$R \sqsubseteq S$	zawieranie ról
$R(b, c)$	asercja wystąpienia roli
s	sygnatura
s_{\max_C}	sygnatura maksymalnego pokrycia dla konceptu C
S_O	zbiór ontologii
S_{KS}	zbiór źródeł wiedzy
S_{M_A}	zbiór odwzorowań atrybutów
S_{M_C}	zbiór odwzorowań konceptów
S_{M_R}	zbiór odwzorowań ról
$S_{M_{s(C)}}$	zbiór odwzorowań sygnatur
\mathcal{Z}	źródło danych
$\forall R.C$	kwantyfikacja ogólna
$\exists R.C$	kwantyfikacja szczegółowa
$\neg C$	negacja konceptu
\top	koncept uniwersalny
\perp	koncept pusty
Δ^I	dziedzina interpretacji
$\{a_1, \dots, a_n\}$	koncept wyliczany
$\geq_n R$	ograniczenie liczebnościowe dolne
$\leq_n R$	ograniczenie liczebnościowe górne

Rozdział 1. Wstęp

1.1 Jak dane stają się wiedzą

Słowo *integracja* (łac. *integratio*) tłumaczy się jako „zespolenie, scalenie, tworzenie całości z części” [TOKA1970]. Znaczenie to przenosi się bezpośrednio, kiedy słowo „integracja” jest używane w dziedzinie informatyki. Jednakże użycie słowa „integracja” implikuje również pytanie o to, co jest integrowane. W tytule tej rozprawy zawarte jest sformułowanie „integracja heterogenicznych źródeł wiedzy”. Zasadnicze pytanie, które należy w tym kontekście zadać, brzmi, czym jest owa wiedza: wiedza w odniesieniu do człowieka i wiedza w odniesieniu do komputera. Oczywiście autor rozprawy nie rości sobie prawa do przeprowadzenia kompletnej analizy postawionego problemu. Nie odnosi się do żadnego z systemów filozoficznych, a jedynie przybliża takie pojęcia, jak wiedza, dane i informacje, często nieświadomie błędnie używane jako pojęcia synonimiczne. Ta krótka analiza ma na celu jedynie przybliżenie Czytelnikowi tematu niniejszej rozprawy i problemów, jakie w tej rozprawie autor próbuje rozstrzygnąć.

Dane (łac. *indicium*), w najbardziej ogólnym systemowym sensie, to „wszystko co jest lub może być przetwarzane umysłowo lub komputerowo” [GADO1999]. W tym sensie dane są pojęciem relatywnym i istnieją tylko razem z pojęciem przetwarzania. Dlatego pojęcie „dane” należy głównie do słownika informatyki i matematyki. Dana jest pewną reprezentacją informacji. Wyobraźmy sobie sytuację z życia. Pani domu przygotowuje listę zakupów, zapisując ją w notesie. Spis zapisany na kartce to dane. Mógłby być on zapisany w komputerze lub zamiast nazw mogłyby być to kody kreskowe wybranych produktów lub odpowiednie fale głosowe i w ten sposób dane uległyby zmianie, jednak nie przekazywana informacja. Ta zostałaby dokładnie taka sama. Możemy wyobrazić sobie również, że ta sama dana może przekazywać również inną informację. Spis produktów żywnościowych stworzony przez gospodynię może nie być listą zakupów, a listą produktów, których należy unikać w stosowanej przez nią diecie. W ten sposób możemy przyjąć, że *informacja* jest pewnym przekazem, który w różny sposób może być reprezentowany jako zbiór danych. **Wiedza** zaś jest pewną informacją, której osoba lub inna jednostka jest świadoma. Rozszerzając nasz przykład, jeśli gospodyni przekaże kartkę z listą produktów swojemu mężowi, to ten, znając kontekst sytuacji i na podstawie własnego doświadczenia, wie, że powinien udać się z powyższą listą do sklepu i zrobić zakupy. Mąż gospodyni staje się jednostką świadomą tych informacji. Zacytujmy kilka definicji wiedzy zaczerpniętych z literatury. Wiedza jest:

- 1) „informacją połączoną z doświadczeniem, kontekstem, interpretacją i refleksją” [DAVE1998];
- 2) „opinią człowieka przechowywaną w ludzkim umyśle, zebraną na drodze doświadczenia i współpracy z innymi ludźmi” [SUNA2002].

O wiedzy mówi się również jako o informacji ocenianej i organizowanej przez ludzki umysł w taki sposób, że może być używana w określonym celu, przykładowo do wyciągania wniosków lub w celu wyjaśniania. W kontekście bardziej informatycznym zaś, o wiedzy mówimy jako o fizycznej, umysłowej lub elektronicznej strukturze zależności pomiędzy rzeczywistymi i abstrakcyjnymi bytami.

To zrozumienie wiedzy z płaszczyzny człowieka przenosimy na płaszczyznę komputera, traktując również komputer jako „jednostkę świadomą” pewnych informacji. Rzeczywiście, w informatyce mówimy o wiedzy, kiedy jest ona zorganizowana w sposób odpowiedni do celu. Przy czym celem tym jest zazwyczaj możliwość wywnioskowania nowych informacji z informacji podanych wprost. Kiedy mówimy o elektronicznej strukturze zależności pomiędzy rzeczywistymi i abstrakcyjnymi bytami, to już w sposób jednoznaczny wprowadzamy formalny opis rzeczywistości zrozumiałej dla komputera. Definicje zaczerpnięte z [SUNA2002] oraz [DAVE1998] przekładają się na zagadnienia poruszane w sztucznej inteligencji, dziedzinie informatyki, której zadaniem jest konstruowanie maszyn i oprogramowania zdolnego rozwiązywać problemy niepoddające się algorytmizacji w sposób efektywny, z wykorzystaniem pewnego modelu wiedzy. Człowiek, posiadając pewien zasób wiedzy oraz określone doświadczenie i nie działając zgodnie z żadnym algorytmem, podejmuje decyzje właśnie wykorzystując tę wiedzę i to doświadczenie.

W niniejszym wstępie wprowadzamy również intuicyjne rozumienie pojęcia „źródło wiedzy”. Formalna definicja zostanie podana w kolejnych rozdziałach rozprawy. Przez **źródło wiedzy** rozumiemy zbiór danych lub informacji podanych w formie zrozumiałej dla odbiorcy, tzn. takich danych i takich informacji, które przez tego odbiorcę mogą być przekształcone w wiedzę. Należy zauważyć, że jeden obiekt może być źródłem wiedzy dla odbiorcy *a* i nie być źródłem wiedzy dla odbiorcy *b*. W świecie rzeczywistym nasuwa się przykład, jakim jest książka. Książka jest źródłem wiedzy tylko dla odbiorcy potrafiącego czytać i będącego na poziomie intelektualnym umożliwiającym zrozumienie informacji w tej książce przekazanych. Kiedy mówimy o integracji źródeł wiedzy, rozumiemy, że system integrujący (którego częścią mogą być również ludzie) scala wiedzę pochodzącą z różnych źródeł w jeden spójny zbiór. Należy pamiętać, że źródłami wiedzy dla systemu integrującego mogą być tylko ściśle określone obiekty, te, które zawierają dane i informacje możliwe do przekształcenia w wiedzę przez system integrujący. W tytule rozprawy mówimy o **źródłach heterogenicznych**, a więc różnorodnych zarówno pod względem formy, jak i treści, ale nie wychodzimy poza obręb źródeł wiedzy właściwych dla danego systemu integrującego.

W tytule tej rozprawy jest również mowa o **logice opisowej** [BAAD2003], [BRAC2004], będącej narzędziem umożliwiającym zbudowanie systemu integrującego. Daje ona podstawy teoretyczne, na których bazuje cała niniejsza rozprawa.

1.2 Od Sieci WWW do Sieci Semantycznej

Sieć WWW, której koncepcję stworzył Tim Berners-Lee w 1989 roku, jest dzisiaj największym elektronicznym źródłem informacji zrozumiałej przez ludzi. Niestety informacje zapisane w sieci WWW nie dają się przetwarzać automatycznie przez komputery. Dlatego Tim Berners-Lee, James Handler i Ora Lassila zaproponowali w 2001 roku wzbogacenie stron WWW o elementy semantyczne rozpoznawane przez komputery i przekształcenie sieci WWW w Sieć Semantyczną (ang. *Semantic Web*) [BERN2001]. Elementy semantyczne, czyli pojęcia używane w danym źródle informacji, są rozpoznawane przez komputery, które docierają po odnośnikach hipertekstowych do dokumentów zawierających definicje tych elementów, czyli do **ontologii**. W zamierzeniach twórców Sieci Semantycznej, umożliwi

to stworzenie programów wykonujących skomplikowane zadania, wymagające rozumienia semantyki dokumentów, które w sieci WWW są wykonywane tylko przez człowieka.

Każdy użytkownik Sieci Semantycznej będzie mógł tworzyć źródła wiedzy wzbogacone o odpowiednie elementy semantyczne, tak jak w sieci WWW każdy użytkownik może tworzyć strony WWW [SWEB2001]. Również każdy użytkownik Sieci Semantycznej będzie mógł definiować własne ontologie, a różne źródła wiedzy będą mogły używać elementów semantycznych zdefiniowanych w różnych ontologiach. Oznacza to, że Sieć Semantyczna tak samo jak sieć WWW będzie miała charakter skrajnie zdecentralizowany, stając się ogromną rozproszoną bazą wiedzy.

1.3 Cele i teza pracy

Potraktowanie Sieci Semantycznej jako rozproszonej bazy wiedzy prowadzi do określenia następujących celów rozprawy:

- 1) Przedstawienie modelu systemu integracji wiedzy.
- 2) Opracowanie algorytmów i metod służących do wytworzenia zamodelowanego systemu.
- 3) Wykazanie, że opracowane algorytmy i metody sprawdzają się w zastosowaniach praktycznych.

Realizacja tych celów służy udowodnieniu następującej tezy rozprawy:

Zaproponowane w rozprawie metody łączenia ontologii i algorytmy przetwarzania zapytań w rozproszonych bazach wiedzy pozwalają na konstrukcję systemu integracji wiedzy o parametrach jakościowych i ilościowych umożliwiającą jego praktycznie zastosowanie we współczesnych systemach opartych na wiedzy.

1.4 Układ rozprawy

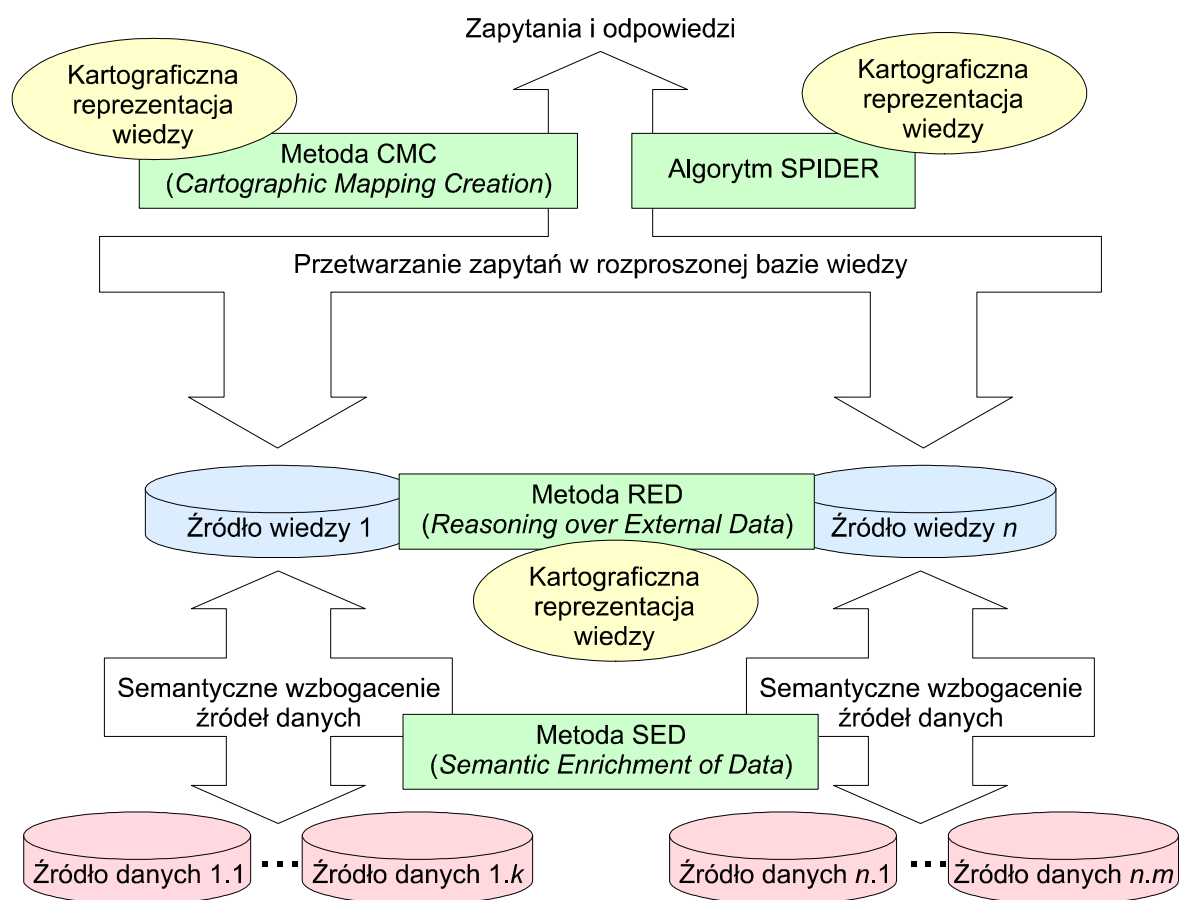
Rozprawa jest podzielona na następujące rozdziały:

Rozdział drugi zawiera aktualny stan wiedzy na temat integracji wiedzy rozproszonej. Wprowadza podstawowe pojęcia dotyczące zagadnień zarządzania wiedzą, w sposób szczególny koncentrując się na zagadnieniach reprezentacji wiedzy w systemach opartych na logice opisowej (ang. *Description Logic*). Rozdział ten opisuje również konstrukcję Sieci Semantycznej, której idea jest głównym motorem rozwoju rozproszonych systemów zarządzania wiedzą. W ostatniej części rozdziału został przedstawiony model systemu integracji wiedzy oparty na logice opisowej.

Rozdziały od trzeciego do ósmego zawierają opis procesu integracji wiedzy w systemie *Knowledge Integration Support System* (KISS) z wykorzystaniem metod łączenia ontologii i algorytmów przetwarzania zapytań w rozproszonych bazach wiedzy, zaproponowanych w tej rozprawie. Przebieg procesu integracji wiedzy w systemie KISS został przedstawiony na rysunku 1.1.

Na tym rysunku białe strzałki pokazują proces znajdowania odpowiedzi na zadane zapytanie w rozproszonym systemie zarządzania wiedzą. Zadane zapytanie jest przetwarzane na zbiór zapytań do źródeł wiedzy, czyli wzbogaconych semantycznie źródeł danych. Seledynowe prostokąty oznaczają metody integracji wiedzy zaproponowane w rozprawie,

wykorzystywane w kolejnych etapach procesu integracji wiedzy. Żółta elipsa oznacza wykorzystanie zaproponowanej w [GOCZ2005] przez W. Waloszkę *kartograficznej reprezentacji wiedzy*.



Rysunek 1.1 Proponowane w rozprawie metody łączenia ontologii i algorytmy przetwarzania zapytań

Rozdział trzeci zawiera opis modelu systemu integracji wiedzy. W tym rozdziale pokazano, jak model systemu integracji ontologii [CALV2001], [CALV2002] można dostosować do praktycznych warunków Sieci Semantycznej.

Rozdział czwarty zawiera wprowadzenie do kartograficznej reprezentacji wiedzy. Rozdział ten obejmuje zagadnienia dotyczące kartograficznego reprezentowania wiedzy terminologicznej i asercjonalnej oraz algorytmów przetwarzania wiedzy reprezentowanej kartograficznie. W tym rozdziale zostało również umieszczone uzasadnienie wyboru metody kartograficznej jako metody reprezentacji wiedzy w systemie KISS.

Rozdział piąty zawiera opis procesu wnioskowania z danych zewnętrznych. W tym rozdziale zostały wprowadzone dwie nowe metody: metoda wnioskowania z danych zewnętrznych RED (*Reasoning over External Data*) [GOCZ2007], oraz metoda semantycznego wzbogacania źródeł SED (*Semantic Enrichment of Data*) [GOCZ2007]. Metoda RED wykorzystuje metodę SED do opisanie kartograficznie danych zewnętrznych. Dodatkowo, metoda RED dostosowuje algorytmy wnioskujące z wiedzy opisanej kartograficznie w taki sposób, aby możliwe było ich zastosowanie, jeśli dane dotyczące osobników nie są bezpośrednio opisane kartograficznie.

Rozdział szósty zawiera opis procesu przetwarzania zapytań w rozproszonej bazie wiedzy. W tym rozdziale opisano nową metodę tworzenia odwzorowań CMC (*Cartographic Mapping Creation*) [GOCZ2008a] pomiędzy **ontologią nadrzędną** – zintegrowanym schematem wiedzy, a **ontologiami podrzędnymi** – schematami wiedzy opisującymi poszczególne źródła wiedzy. Metoda ta jest używana przez zaproponowany w rozprawie algorytm SPIDER [GOCZ2005a], [GOCZ2008a], odpowiedzialny za przekształcanie zapytań z terminów zdefiniowanych w ontologii globalnej w terminy zdefiniowane w ontologiach lokalnych oraz łączenie odpowiedzi z różnych źródeł wiedzy w jedną spójną odpowiedź.

Rozdział siódmy zawiera opis systemu KISS (*Knowledge Inference Support System*), w którym zastosowano metody integracji wiedzy zaproponowane w rozprawie. W tym rozdziale zostały zawarte cele systemu KISS oraz rola systemu KISS w procesie integracji wiedzy.

Rozdział ósmy zawiera ocenę metod zaproponowanych w rozprawie, poprzez pokazanie praktycznego zastosowania systemu KISS. W rozdziale opisano wykorzystanie systemu KISS w rzeczywistych warunkach poważnego przedsięwzięcia informatycznego.

Rozdział dziewiąty zawiera wnioski z prac opisanych w tej rozprawie. Również w tym rozdziale zostały opisane dalsze kierunki badań.

1.5 Konwencje notacyjne

Metody zaproponowane w rozprawie bazują na logice opisowej. Logika opisowa jest formalizmem opisu wiedzy i ma ściśle określoną konwencję zapisu [BAAD2003]. Taką konwencję zapisu przyjęto również w tej rozprawie.

Ponadto w rozprawie przyjęto, że:

- 1) Terminy pojawiające się w rozprawie po raz pierwszy są zapisane pogrubioną kursywą, np.: **ontologia globalna**.
- 2) Wszystkie terminy obcojęzyczne pisane są kursywą niepogrubioną, np.: *integratio*.
- 3) Fragmenty kodów lub dokumentów XML są pisane czcionką maszynową; np.: `<nazwa>`.
- 4) Pozostałe wyróżnione fragmenty tekstu pisane są czcionką podkreśloną, np.: ważna cecha.

Rozdział 2. Wprowadzenie w zagadnienia integracji wiedzy w Sieci Semantycznej

2.1 Logika opisowa jako formalny sposób zapisu ontologii

Logika opisowa (ang. *Description Logic* – DL) jest formalnym językiem reprezentacji wiedzy [BAAD2003]. W rzeczywistości jednak mówimy o logikach opisowych (ang. *Description Logics* – DLs) jako rodzinie formalizmów, które w pierwszej kolejności reprezentują za pomocą pewnych pojęć i zależności między nimi wiedzę z danej dziedziny – **terminologię** (ang. *terminology*), a następnie, wykorzystując te pojęcia, określają cechy **osobników** (ang. *individuals*) z tej dziedziny – **opis świata** (ang. *world description*).

Terminologia

Terminologia (inaczej **TBox**) składa się z:

- 1) **konceptów** (ang. *concepts*) oznaczających zbiór osobników,
- 2) **ról** (ang. *roles*) będących binarnymi relacjami pomiędzy osobnikami.

Koncepty i role mogą mieć przypisane nazwy. W takim przypadku mówimy, że są to koncepty i role atomowe (nazwy konceptów i ról). Koncepty i role atomowe pojawiające się w terminologii możemy podzielić na dwa zbiory: zbiór **symboli nazwanych** (ang. *name symbols*) i zbiór **symboli bazowych** (ang. *base symbols*). Symbole bazowe pojawiają się tylko po prawej stronie aksjomatów, a symbole nazwane również po lewej. Symbole nazwane często określamy jako **koncepty i role definiowane** (ang. *defined concepts and roles*), a symbole bazowe jako **koncepty i role pierwotne** (ang. *primitive concepts and roles*). Logiki opisowe między sobą różnią się zbiorem **konstruktorów** (ang. *constructs*) które możemy wykorzystać do tworzenia konceptów i ról definiowanych. Zbiór takich konstruktorów w logikach opisowych nazwany jest **językiem opisu** (ang. *description language*).

Minimalny praktycznie użyteczny zbiór konstruktorów został wprowadzony w języku \mathcal{ALC} ¹. Koncepty złożone mogą być w języku \mathcal{ALC} tworzone za pomocą następujących konstruktorów:

$C, D \rightarrow$	A		koncept atomowy
	\top		koncept uniwersalny
	\perp		koncept pusty
	$\neg C$		negacja konceptu
	$C \sqcap D$		przecięcie konceptów
	$C \sqcup D$		suma konceptów

¹ Język \mathcal{ALC} jest równoważny językowi \mathcal{ALUE} , ponieważ suma konceptów i kwantyfikacja szczegółowa mogą być wyrażone przez negację konceptu: $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ i $\exists R.C \equiv \neg \forall R. \neg C$.

$$\begin{array}{ll} \forall R.C & | \quad \text{kwantyfikacja ogólna} \\ \exists R.C & | \quad \text{kwantyfikacja szczegółowa} \end{array}$$

Literą A oznaczamy koncepty atomowe, literami C i D koncepty złożone. Literą R oznaczamy role atomowe. Intuicyjnie poprzez koncept uniwersalny rozumiemy zbiór wszystkich osobników należących do dziedziny wiedzy, a poprzez koncept pusty zbiór pusty osobników. Aby opisać znaczenie kwantyfikacji ogólnej i szczegółowej, wprowadźmy jeszcze dwa pojęcia: **podmiotu** i **dopełnienia roli** (ang. *role subject* i *role filler*). Jeśli mówimy, że dwa osobniki x i y pozostają ze sobą w relacji binarnej R_1 , co zapisujemy $R_1(x, y)$, to x nazywamy podmiotem roli, a y jej dopełnieniem. Koncept zdefiniowany jako kwantyfikacja ogólna jest to zbiór takich osobników, które jako podmioty roli pozostają w relacji R wyłącznie z osobnikami należącymi do konceptu C . Kwantyfikacja szczegółowa oznacza zbiór tych osobników, które jako podmioty roli pozostają w relacji R z przynajmniej jednym osobnikiem będącym wystąpieniem konceptu C .

Aby znaczenie konstrukcji w języku \mathcal{ALC} wprowadzić formalnie, rozważmy pojęcie **interpretacji** (ang. *interpretation*). Interpretacja \mathcal{I} składa się z niepustego zbioru $\Delta^{\mathcal{I}}$ nazwanego **dziedziną interpretacji** i z **funkcji interpretacji**, która każdemu atomowemu konceptowi A przypisuje zbiór $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, a każdej roli atomowej R binarną relację $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Funkcja interpretacji jest rozszerzona w języku \mathcal{ALC} poprzez następujące definicje, pokazujące jak można tworzyć koncepty definiowane.

Definicja 2.1 Funkcja interpretacji w języku \mathcal{ALC}

$$\begin{array}{ll} \top^{\mathcal{I}} & = \Delta^{\mathcal{I}} \\ \perp^{\mathcal{I}} & = \emptyset \\ (\neg C)^{\mathcal{I}} & = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} & = C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (C \sqcup D)^{\mathcal{I}} & = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} & = \{a \in \Delta^{\mathcal{I}} : (\forall b \in \Delta^{\mathcal{I}} ((a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}))\} \\ (\exists R.C)^{\mathcal{I}} & = \{a \in \Delta^{\mathcal{I}} : (\exists b \in \Delta^{\mathcal{I}} ((a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}))\} \end{array}$$

W następnej kolejności należy wprowadzić aksjomaty terminologiczne, które pozwalają definiować zależności między konceptami i między rolami. W najbardziej ogólnej formie aksjomaty mają następującą postać:

Definicja 2.2 Aksjomaty terminologiczne

- (1) $C \sqsubseteq D$ ($R \sqsubseteq S$) oraz
 - (2) $C \equiv D$ ($R \equiv S$)
- gdzie C i D są to koncepty, a R i S role.
-

Aksjomaty pierwszego typu (1) nazywamy **aksjomatami zawierania** (ang. *inclusions*), a drugiego typu (2) **aksjomatami równości** (ang. *equalities*). Znaczenie powyższych aksjomatów jest takie, jakiego się spodziewamy. Tak więc interpretacja \mathcal{I} **spełnia** (ang. *satisfies*) aksjomat zawierania $C \sqsubseteq D$, jeśli $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, a aksjomat równości $C \equiv D$, jeśli $C^{\mathcal{I}} = D^{\mathcal{I}}$. Jeśli interpretacja \mathcal{I} spełnia aksjomat (zbiór aksjomatów), to mówimy, że jest ona **modelem** tego aksjomatu (zbioru aksjomatów).

Opis świata

Opis świata (inaczej **ABox**) obejmuje osobniki, którym nadajemy nazwy oraz tworzymy asercje opisujące cechy tych osobników.

Definicja 2.3 Asercje opisujące cechy osobników

(1) $C(a)$ oraz

(2) $R(b, c)$

gdzie C jest konceptem, a R rolą.

Asercje mogą być dwojakiego rodzaju. Pierwsza z asercji (1) to **asercja wystąpienia konceptu** (ang. *concept assertion*), a druga (2) to **asercja wystąpienia roli** (ang. *role assertion*). Asercja wystąpienia konceptu stwierdza, że osobnik a zawiera się w interpretacji konceptu C , zaś asercja wystąpienia roli stwierdza, że osobnik c jest dopełnieniem roli R dla osobnika b . Znaczenie powyższych asercji jest następująca: interpretacja \mathcal{I} **spełnia** asercję wystąpienia konceptu $C(a)$, jeżeli $a^{\mathcal{I}} \in C^{\mathcal{I}}$, oraz interpretacja \mathcal{I} spełnia asercję wystąpienia roli $R(b, c)$ jeżeli $(b, c)^{\mathcal{I}} \in R^{\mathcal{I}}$. Jeśli interpretacja \mathcal{I} spełnia asercję (zbiór asercji), to mówimy, że jest ona **modelem** tej asercji (zbioru asercji).

Interpretacja \mathcal{I} jest **modelem ontologii**, jeżeli jest modelem zbioru aksjomatów zdefiniowanych w terminologii i jest modelem zbioru asercji zdefiniowanych w opisie świata.

Przykład 2.1 Terminologia i opis świata

Dla zobrazowania terminologii i opisu świata posłużmy się przykładem. Wyobraźmy sobie, że dziedziną, którą chcemy opisać, jest rodzina. Zdefiniujmy dla rodziny pewną terminologię:

- (1) $Osoba \equiv Kobieta \sqcup Mężczyzna$
- (2) $\perp \equiv Kobieta \sqcap Mężczyzna$
- (3) $Rodzic \equiv Osoba \sqcap \exists maDziecko.Osoba$
- (4) $Ojciec \equiv Mężczyzna \sqcap Rodzic$
- (5) $Matka \equiv Kobieta \sqcap Rodzic$

W tej terminologii koncepty atomowe to *Kobieta*, *Mężczyzna*, *Osoba*, *Rodzic*, *Ojciec* i *Matka*. Wśród nich koncepty pierwotne to *Kobieta* i *Mężczyzna* (nie występują po lewej stronie aksjomatów). Pozostałe koncepty atomowe to koncepty definiowane. Terminologia ta stwierdza, że każda kobieta i każdy mężczyzna jest osobą, a osoba może być albo mężczyzną albo kobietą (aksjomat (1)). Dodatkowo, żadna kobieta nie jest mężczyzną i żaden mężczyzna nie jest kobietą (aksjomat (2)). Wiemy również, że osoba może mieć dziecko, a każda osoba, która ma przynajmniej jedno dziecko będące osobą, jest rodzicem (aksjomat (3)). Pozostaje jeszcze zdefiniować, kim jest matka i ojciec. Matka to kobieta będąca rodzicem (aksjomat (5)), zaś ojciec to mężczyzna będący rodzicem (aksjomat (4)).

Posługując się konceptami i rolami zdefiniowanymi w tej terminologii, wygenerujemy przykładowy opis świata:

Kobieta(Anna)

Kobieta(Joanna)

Mężczyzna(Karol)

maDziecko(Anna, Joanna)

W ten sposób stwierdzamy, że *Anna* i *Joanna* to kobiety, *Karol* to mężczyzna, a *Anna* ma dziecko, którym jest *Joanna*.

2.2 Ontologie a logika opisowa

Czym jest logika opisowa w odniesieniu do kluczowego terminu w dziedzinie zarządzania wiedzą, tj. ontologii? Logika opisowa jest językiem zapisu ontologii. Czym więc jest sama ontologia? Najbardziej popularna definicja [GRUB1993] mówi, że „Ontologia jest bezpośrednią specyfikacją konceptualizacji”. Konceptualizacja w tym kontekście oznacza abstrakcyjny model opisujący pewien aspekt świata, model będący zbiorem definicji własności konceptów i zależności między nimi. Stwierdzenie „bezpośrednia specyfikacja” oznacza, że model powinien być określony w jednoznacznym, formalnym języku umożliwiającym zrozumienie tego modelu przez ludzi i przetwarzanie przez komputery. Kiedy mówimy o przetwarzaniu ontologii, kojarzymy pojęcie przetwarzania z definicją wiedzy wprowadzoną we wstępie do tej rozprawy. Zauważmy, że ontologia, aby stała się wiedzą w rzeczywistym tego słowa znaczeniu, musi posiadać świadomego odbiorcę, który będzie potrafił informacje zapisane w ontologii odpowiednio przetworzyć. Mówimy w tym przypadku o informacjach (nie o danych), ponieważ ontologia nadaje danym ściśle określoną semantykę. Kolejny podrozdział wprowadza pojęcie systemu reprezentacji wiedzy jako „jednostki świadomej” wiedzy zapisanej w ontologiach. Poczynając od tego podrozdziału, w całej rozprawie przyjęto, że jeśli mówimy o ontologiach, to mówimy o abstrakcyjnym modelu zdefiniowanym w terminach logiki opisowej.

2.3 Wnioskowanie i systemy reprezentacji wiedzy

Użycie ontologii wymaga nie tylko stworzenia odpowiedniego modelu opisującego pewien aspekt świata, ale również narzędzi umiających przetwarzać informacje zapisane w ontologiach. Jeśli mówimy o przetwarzaniu, to mamy na myśli przede wszystkim umiejętność wnioskowania, czyli odkrywania nowych faktów, niepodanych w ontologiach wprost.

W logice opisowej definiujemy dwa podstawowe typy problemów wnioskowania. Pierwszy z nich to **wnioskowanie z terminologii**, drugi zaś to **wnioskowanie o osobnikach**. I tak do podstawowych problemów wnioskowania z terminologii należą:

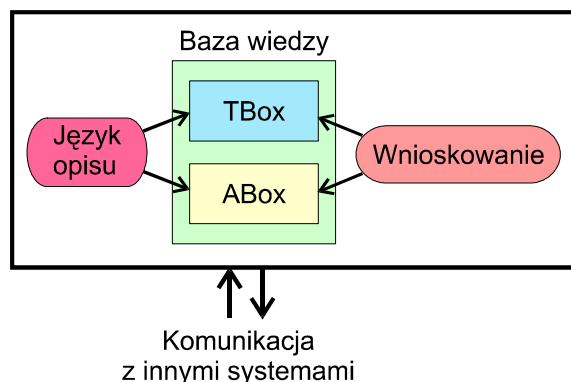
- 1) **problem zawierania** (ang. *subsumption problem*) – czy zbiór wystąpień konceptu C jest zawsze podzbiorem wystąpień konceptu D ? (Czy $C \sqsubseteq D$?);
- 2) **problem spełnialności** (ang. *satisfiability problem*) – czy koncept C może mieć wystąpienia (Czy $C \sqsubseteq \perp$?);
- 3) **problem równoważności** (ang. *equivalence problem*) – czy zbiory wystąpień konceptów C i D są zawsze równe? (Czy $C \sqsubseteq D$ i $D \sqsubseteq C$);
- 4) **problem rozłączności** (ang. *disjointness problem*) – czy zbiory konceptów C i D są zawsze rozłączne? (Czy $C \sqcap D \sqsubseteq \perp$);

Warto zauważyć, że każdy z tych problemów sprowadza się do problemu zawierania: $C \sqsubseteq \perp \leftrightarrow C \sqsubseteq \perp$, $C \sqsubseteq D \leftrightarrow C \sqsubseteq D \wedge D \sqsubseteq C$, $C \sqcap D \sqsubseteq \perp \leftrightarrow C \sqsubseteq \neg D$. Do problemów wnioskowania o osobnikach należą:

- 1) **problem określenia zbioru wystąpień konceptu** (ang. *retrieval problem*) – jakie osobniki należą do konceptu C ?
- 2) **problem sprawdzenia przynależności** (ang. *instance check problem*) – czy dany osobnik należy do konceptu C ?
- 3) **problem sprawdzenia spójności** (ang. *consistency check*) – czy istnieje niepusty model ontologii?

Wymienione tutaj problemy wnioskowania należą do najbardziej powszechnych. Zdefiniowano również problemy bardziej złożone, do których należy między innymi zapytanie o wiedzę potrzebną do wywnioskowania pewnego faktu [GOCZ2006e].

Najbardziej popularne systemy umożliwiające przeprowadzenie procesu wnioskowania to **systemy reprezentacji wiedzy** (ang. *knowledge representation systems*). Definicja formalna [BAAD2003] systemu reprezentacji wiedzy opartego na logice opisowej mówi, że są to „systemy dostarczające takie usługi jak: definiowanie bazy wiedzy, wnioskowanie z zawartości bazy wiedzy i manipulowanie tą zawartością”. Poprzez bazę wiedzy rozumiemy terminologię i opis świata. Rysunek 2.1 pokazuje ogólną architekturę systemu reprezentacji wiedzy. Terminologia (TBox) i opis świata (ABox) są opisane pewnym językiem opisu o zadanej ekspresywności. Wnioskowanie jest prowadzone na podstawie danych zapisanych w terminologii i opisie świata. Z systemem reprezentacji wiedzy komunikują się inne systemy, ponieważ systemy reprezentacji wiedzy są w ogólności zanurzone w szerszym środowisku. Inne systemy komunikują się z systemem reprezentacji wiedzy, zadając zapytania do bazy wiedzy lub ją modyfikując.



Rysunek 2.1. Architektura systemu reprezentacji wiedzy opartego na logice opisowej [BAAD2003]

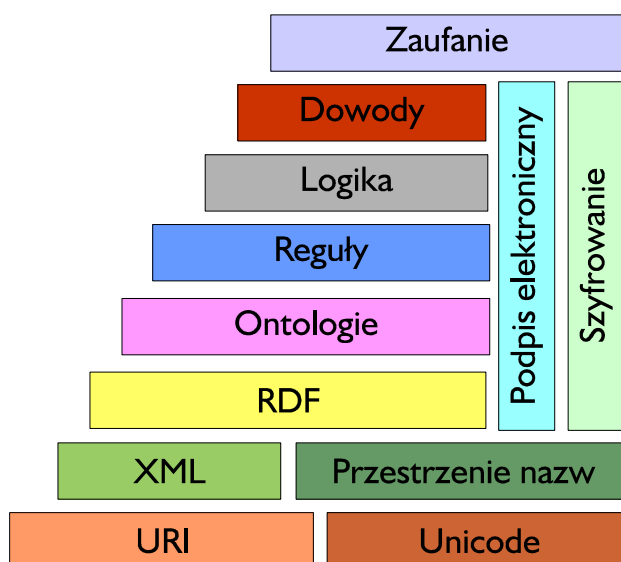
System reprezentacji wiedzy jest również niezbędną częścią systemu integracji wiedzy. Kolejne dwa podrozdziały zawierają odpowiedzi na pytania, czym jest integracja wiedzy i jaki jest model systemu integracji wiedzy.

2.4 Integracja w Sieci Semantycznej

Wraz z rozwojem Internetu opracowanie metod zarządzania wiedzą poprzez systemy integracji wiedzy stało się jednym z fundamentalnych problemów zarządzania informacją. Aktualnie istniejące systemy wyszukiwania informacji w Internecie stają się coraz mniej wystarczające. W tym obszarze powstają liczne inicjatywy stawiające sobie za cel usystematyzowanie wiedzy zapisanej w źródłach internetowych. Wśród nich najbardziej znana jest inicjatywa Sieci Semantycznej. „Sieć Semantyczna jest to rozszerzenie istniejącej sieci WWW o mechanizmy semantyczne, tak aby informacje dostępne w tej sieci były dobrze zdefiniowane i umożliwiały lepszą współpracę komputerom i ludziom” [BERN2001]. Zanim wprowadzimy pojęcie integracji w Sieci Semantycznej, przyjrzyjmy się bliżej, czym taka sieć jest [GOCZ2006a].

Sieć Semantyczna

Logiczną konstrukcję Sieci Semantycznej bardzo dobrze ilustruje tzw. „torcik” Sieci Semantycznej przedstawiony na rysunku 2.2. Dolną warstwę Sieci, jej podstawę, stanowi system jednoznacznego adresowania zasobów Sieci identyfikatorami URI (*Uniform Resource Identifier*) [URIRFC2396]. Nie wystarcza posługiwanie się adresami typu URL (*Uniform Resource Locator*), gdyż pod jednym adresem URL może znajdować się wiele zasobów. Drugim elementem podstawy jest powszechny alfabet Unicode, pozwalający na binarne kodowanie znaków dowolnego alfabetu.



Rysunek 2.2. Torcik Sieci Semantycznej [SWEB2001]

Drugą warstwę torcika stanowi baza syntaktyczna, jaką jest język XML (*Extensible Markup Language*) [XML2006]. Na tym języku oparto składnię pozostałych warstw, w szczególności warstw semantycznych Sieci. Warstwa XML umożliwia definiowanie typów dokumentów (XML Schema [XMLSCHEMA2004]), co stwarza możliwości lepszego strukturalizowania, a co za tym idzie – uporządkowania zasobów Sieci. Istotnym elementem tej warstwy jest też możliwość definiowania przestrzeni nazw (ang. *namespaces*), co pozwala na unikanie konfliktów w sytuacji, gdy w różnych miejscach Sieci pod tymi samymi nazwami rozumie się różne pojęcia. Warstwa XML stwarza też możliwości wyrażania znaczenia treści dokumentów poprzez stosowanie semantycznych (a nie, jak w dokumentach HTML [HTML1999], czysto edytorskich) znaczników w dokumentach Sieci i poprzez strukturalizowanie tych dokumentów w formy drzewiaste, o zagnieżdżających się elementach. Stąd też dokumenty XML nazywa się czasami dokumentami samoopisującymi się (ang. *self-describing documents*). Kolejna, trzecia warstwa Sieci Semantycznej, to warstwa RDF [RDF2004], [MCBR2004] umożliwiająca tworzenie zdań o zasobach identyfikowanych poprzez pewne URI. Przykład takiego zdania został przedstawiony poniżej:

```
<rdf:resource rdf:about="http://pg#JK" pg:nazwisko="Kowalski">
  <pg:stronaDomowa rdf:resource="http://www.pg/page.html"/>
</rdf:resource>
```

Opisywany jest tu – w postaci dwóch trójek RDF – zasób o identyfikatorze `http://pg#JK`. Ma on właściwość `nazwisko` o wartości literalnej `Kowalski` oraz właściwość `stronaDomowa`, która sama jest zasobem o nazwie `http://www.pg/page.html` (i oczywiście

też może mieć swoje właściwości). Właściwości `nazwisko` i `stronaDomowa` pochodzą z pewnej przestrzeni nazw `pg` (i tam też pewnie opisane jest ich znaczenie), natomiast właściwości `resource` i `about` należą do przestrzeni nazw `rdf`, w której zdefiniowano syntaktykę i semantykę RDF. RDF umożliwia definiowanie typów zasobów (służy do tego język RDF Schema [RDFS2004]). Można na przykład zdefiniować typ zasobu `Osoba`, który ma właściwości `nazwisko`, `adres` itd. określonego typu. Umożliwia to budowanie prostych ontologii, w których występują koncepty – typy – i ich wystąpienia – zasoby.

Na kolejnym poziomie „torcika” znajdujemy ontologie bardziej ekspresywne niż zbiory trójek RDF, w szczególności ontologie zapisane w języku OWL-DL [OWL2004], [ANTO2004]. Język ten jest standardem zapisu ontologii w Sieci Semantycznej. Składnia języka OWL jest oparta na trójkach RDF, a jego semantyka oparta jest na logice opisowej, a ściślej na języku opisu o nazwie *SHION*. Język *SHION* obejmuje:

- 1) zbiór konstruktorów języka \mathcal{ALCR}^+ [HRR1999] (język \mathcal{ALC} rozszerzony o konstruktor domknięcia przechodniego roli R^+), oznaczony literą \mathcal{S} ,
- 2) aksjomat hierarchii ról ($R \sqsubseteq S$) oznaczony literą \mathcal{H} ,
- 3) konstruktor *inwersji roli* R^- oznaczony literą \mathcal{I} ,
- 4) konstruktor *konceptu wyliczanego* $\{a_1, \dots, a_n\}$ oznaczony literą \mathcal{O} ,
- 5) konstruktory *niekwalifikowanych ograniczeń liczebnościowych* $\geq n R$, $\leq n R$ oznaczone literą \mathcal{N} oraz
- 6) *dziedziny konkretne* [BAAD1999].

Definicja 2.4 Funkcja interpretacji w języku *SHION*

Funkcja interpretacji języka *SHION* jest funkcją interpretacji języka \mathcal{ALC} rozszerzoną poprzez następujące definicje:

$$(R^+)^{\mathcal{I}} = \{\cup_{i>1} (R^{\mathcal{I}})^i\} \quad (1)$$

$$(R)^{\mathcal{I}} = \{(b, a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} : (a, b) \in R^{\mathcal{I}}\} \quad (2)$$

$$(\{a_1, \dots, a_n\})^{\mathcal{I}} = \{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\} \quad (3)$$

$$(\geq n R)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : (a, b) \in R^{\mathcal{I}}\}| \geq n\} \quad (4)$$

$$(\leq n R)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : (a, b) \in R^{\mathcal{I}}\}| \leq n\} \quad (5)$$

W języku *SHION* istnieje konstruktor domknięcia przechodniego roli. W języku OWL taki konstruktor nie istnieje. Rola jest bezpośrednio definiowana jako przechodnia lub nieprzechodnia. Język *SHION* obejmuje również dziedziny konkretne. O dziedzinach konkretnych mówimy wtedy, kiedy chcemy tworzyć opisy konceptów odnoszące się do konkretnych wartości liczbowych lub tekstowych, jak w poniższym przykładzie:

Przykład 2.2 Zastosowanie dziedzin konkretnych

$CzłowiekDorosły \equiv Człowiek \sqcap \exists maWiek. \geq_{18}$

$CzłowiekWysoki \equiv Człowiek \sqcap \exists maWzrost. \geq_{170}$

Definicja 2.5 Dziedzina konkretna \mathcal{D}

Dziedziną konkretną \mathcal{D} nazywamy parę $\mathcal{D} = (\Delta^{\mathcal{D}}, pred(\mathcal{D}))$, gdzie zbiór $\Delta^{\mathcal{D}}$ reprezentuje dziedzinę, a zbiór $pred(\mathcal{D})$ składa się z nazw predykatów operujących na tej dziedzinie. Z każdą nazwą predykatu $P \in pred(\mathcal{D})$ jest skojarzona krotność n oraz dla każdego n -arnego predykatu $P^{\mathcal{D}}$ zachodzi relacja $P^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$.

Jako przykład dziedziny konkretnej możemy podać dziedzinę konkretną \mathcal{N} , w której dziedziną jest zbiór liczb naturalnych \mathbb{N} , zaś zbiór $pred(\mathcal{N})$ składa się z nazw predykatów binarnych $<, \leq, >, \geq$ oraz nazw predykatów unarnych $<_n, \leq_n, >_n, \geq_n$, gdzie $n \in \mathbb{N}$. To właśnie dziedzina konkretna \mathcal{N} została wykorzystana do zdefiniowania conceptów *CzłowiekDorosły* i *CzłowiekWysoki* w przykładzie 2.2.

W języku OWL poza rolami (ang. *object type property*) występują również atrybuty (ang. *data type property*). Atrybut jest relacją binarną pomiędzy osobnikiem i wartościami należącymi do pewnych dziedzin. Każdy atrybut ma zdefiniowany typ określający zbiór wartości, jakie może przyjmować. W definicji conceptów *CzłowiekDorosły* i *CzłowiekWysoki* z przykładu 2.2 wykorzystano atrybuty *maWiek* i *maWzrost*. Typem obu atrybutów jest zbiór liczb naturalnych \mathbb{N} , czyli dziedzina dziedziny konkretnej \mathcal{N} .

Następna warstwa „torcika” to reguły. Niestety, ekspresywność logiki opisowej w Sieci Semantycznej jest niewystarczająca do opisu występujących w rzeczywistym świecie złożonych zależności pomiędzy osobnikami. Dlatego w ramach inicjatywy Sieci Semantycznej zaproponowano również inną reprezentację wiedzy, znaną i stosowaną od dawna z uwagi na efektywne algorytmy wnioskowania, czyli regułowe metody reprezentacji wiedzy [RUSS2003]. Wśród nich najbardziej popularne bazują na klauzulach Horna. Pomimo tego, że istnieje pewien wspólny obszar semantyczny zarówno dla reguł, jak i logiki opisowej, to istnieją pewne specyficzne konstrukcje możliwe do wyrażenia tylko jedną z wymienionych metod. I tak, istnieją poważne ograniczenia w wyrażaniu w regułach takich konstruktorów logicznych jak negacja, dysjunkcja czy kwantyfikacja egzystencjalna. Z drugiej strony w logikach opisowych nie potrafimy wyrażać takich reguł, w których liczba zmiennych jest większa niż jeden. W Sieci Semantycznej zaproponowano język SWRL (*Semantic Web Rule Language*) [SWRL2004] wzbogacający język OWL o reguły Horna.

Kolejne warstwy Sieci Semantycznej – logika i dowody – obejmują ontologie wyrażone za pomocą mechanizmów ekspresji dostarczanych przez warstwy niższe (logika) oraz moduły wnioskujące (dowody). Nad wszystkimi warstwami Sieci Semantycznej „czuwa” warstwa zaufania. Jest to niezwykle ważna warstwa, gdyż użytkownicy Sieci Semantycznej powinni ufać składającym się na nią źródłom wiedzy, a przynajmniej zdawać sobie sprawę ze stopnia zaufania, jakim mogą obdarzyć dane źródło wiedzy i – tym samym – ontologie prezentowane w Sieci przez to źródło. Jest kilka środków zwiększania takiego zaufania. Należą do nich powszechne stosowanie podpisu cyfrowego oraz szyfrowanie informacji przekazywanej przez Sieć Semantyczną. Istotne jest bowiem, by na etapie wymiany informacji pomiędzy źródłami wiedzy oraz podczas przekazywania informacji ze źródła wiedzy do użytkownika informacja nie ulegała przekłamaniu, celowemu sfalszowaniu i innym zakłóceniom.

Ważną rolę w Sieci Semantycznej odgrywają agenty (ang. *agents*) [JENN2001], rozumiane jako autonomicznie działające programy komputerowe, które potrafią samodzielnie – komunikując się w razie potrzeby z innymi agentami – rozwiązać zadanie postawione im przez człowieka. O ile ontologie stanowią statyczną część Sieci Semantycznej, to jej dynamiczną składową wyznaczają właśnie agenty. Każdy użytkownik Sieci Semantycznej dysponuje zbiorem agentów, wyspecjalizowanych w określonych dziedzinach. Przykładowo: jeden agent może zajmować się sprawami żywieniowymi, poszukując informacji o restauracjach oferujących menu odpowiadające gustom „właściciela” tego agenta, inny może poszukiwać w sieci interesujących wydarzeń kulturalnych, inny jeszcze może nadzorować kwestie zdrowotne użytkownika, komunikując się w tym celu z odpowiednią usługą WWW, itd. Wyobraźnia może podsuwać kolejne dziedziny i zastosowania. Aby sprostać tym zadaniom, agenty nie tylko poszukują w Sieci Semantycznej istotnych dokumentów, ale także korzystają z aktywnych usług WWW opisanych stosownymi ontologiami (zapisanymi w dialekcie OWL-DL o nazwie OWL-S

[OWLS2004]), a także z usług systemów wnioskujących, aby za ich pomocą uzyskać dodatkowe informacje, niedostępne bezpośrednio.

Miejsce integracji wiedzy w Sieci Semantycznej

Dotychczas, mimo że opisaliśmy całą logiczną konstrukcję Sieci Semantycznej, nie wspomnieliśmy o integracji wiedzy i jej miejscu w Sieci Semantycznej. Możemy jednak na podstawie logicznej konstrukcji Sieci Semantycznej wymienić kilka cech Sieci Semantycznej, które bezpośrednio wskazują na potrzebę integracji wiedzy:

- (1) Wielość ontologii często obejmujących tę samą dziedzinę wiedzy. Istotnym elementem umożliwiającym konstruowanie ontologii w Sieci Semantycznej są edytory ontologii, a więc systemy, przy użyciu których powstają ontologie. Ontologie takie są tworzone w wielu miejscach na świecie, przez niezależne grupy. Często powstają ontologie o różnej terminologii, jednak dotyczące tej samej dziedziny wiedzy. Których z tych ontologii mają używać agenty? Czy agenty powinny korzystać tylko z jednej ontologii i tracić informacje zapisane w innych ontologiach? A może agenty powinny korzystać ze wszystkich ontologii dotyczących interesującego dla nich tematu? Jeśli tak, to w jaki sposób mają łączyć wiedzę wydobytą z tych ontologii?
- (2) Brak identyfikowalności zasobów. W Sieci Semantycznej zasoby są jednoznacznie identyfikowane za pomocą URI. Jednak Sieć jest zbyt rozległa, aby było możliwe kontrolowanie nadawania URI. URI może nadać każdy użytkownik Sieci, a tym samym te same semantycznie zasoby mogą mieć w Sieci różne URI. Czy jest więc możliwe, aby agenty rozpoznawały, że wiedza zdobyta w różny sposób w Sieci jest w rzeczywistości wiedzą na temat tego samego zasobu?
- (3) Potrzeba łączenia wiedzy pochodzącej z różnych źródeł w Sieci. Wiedza może pochodzić z różnych źródeł. Często jest to wiedza już uzyskana na drodze wnioskowania. Czy możliwe jest uzupełnianie wiedzy pochodzącej z jednego źródła o wiedzę pochodzącą z innego źródła?

W odniesieniu do problemu (1) mówimy o integracji ontologii (ang. *ontology integration*) [PINT1999]. Poprzez integrację ontologii rozumiemy tworzenie nowej ontologii z wykorzystaniem ontologii już istniejących. W ten sposób agenty powinny widzieć zawsze tylko jedną interesującą ich ontologię, która w rzeczywistości może składać się z wielu ontologii. Mówiąc tutaj o integracji ontologii, w sposób szczególny odnosimy się do problemu integracji terminologii. Jeśli terminologie są zintegrowane poprawnie, to integracja opisów świata sprowadza się w praktyce do identyfikowalności tych samych zasobów, a więc problemu (2), o którym później. Proces integrowania terminologii polega na tworzeniu nowej terminologii z terminologii istniejących (terminologii integrowanych) w taki sposób, że:

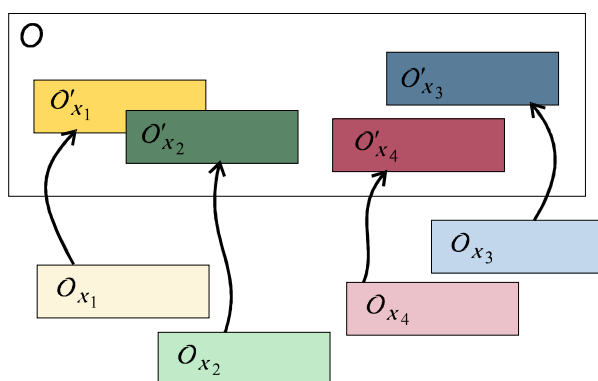
- 1) terminologie integrowane są dostosowywane do konkretnych potrzeb, a następnie
- 2) terminologie te są wzbogacane o nowe terminy (koncepty, role) i zależności między nimi.

Terminologie integrowane mogą być użyte bezpośrednio, jeśli są odpowiednie dla danego zastosowania. Dostosowanie zaś terminologii integrowanych do indywidualnych potrzeb może albo specjalizować koncepty, role lub atrybuty, albo je wzbogacać. O specjalizacji mówimy, jeśli terminy pojawiające się w terminologii są zbyt ogólne; np. potrzebujemy wiedzy na temat pieczywa (koncept *Pieczywo*), a w terminologii integrowanej zamodelowano wiedzę na temat produktów zbożowych (koncept *ProduktZbożowy*). W ten sposób w odniesieniu do konceptu *ProduktZbożowy* mówimy o jego specjalizacji. Poprzez wzbogacanie rozumiemy sytuację odwrotną: w terminologii integrowanej mamy

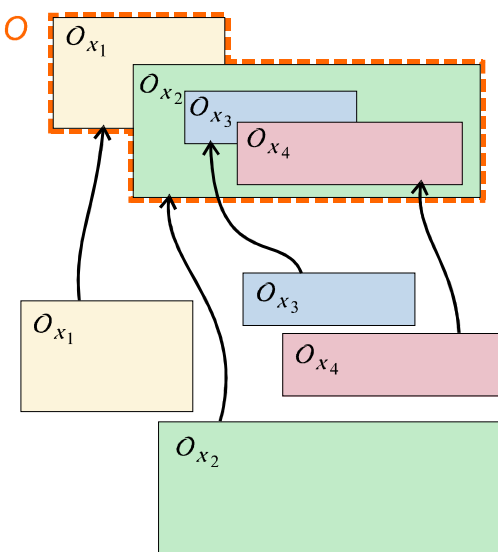
zdefiniowany koncept *Pieczyno*, a potrzebujemy konceptu *ProduktZbozowy*. W ten sposób wzbogacamy koncept *Pieczyno*. W procesie transformacji terminologii pewne terminy mogą ulegać specjalizacji, a pewne wzbogacaniu. A zatem w rzeczywistości mówimy nie o specjalizacji lub wzbogacaniu terminologii, a o specjalizacji lub wzbogacaniu terminów pojawiających się w terminologii. W wyjątkowych sytuacjach, kiedy część terminów ulega specjalizacji, a pozostałe terminy pozostają bez zmian, możemy mówić o specjalizacji terminologii. Analogiczna sytuacja zachodzi w przypadku wzbogacania terminologii.

W drugim kroku procesu integracji w terminologii zintegrowanej (wynikowej) definiowane są dodatkowe role i koncepty potrzebne w konkretnym zastosowaniu. Przykładowo: założmy, że jedna z terminologii integrowanych zawiera koncept *Pieczyno*, a inna koncept *Lek*. Terminologia wynikowa ma opisywać interakcje pomiędzy lekami a pieczywem. W takiej sytuacji w terminologii wynikowej dodatkowo potrzebna jest rola *wchodziWInterakcję*.

a)



b)



Rysunek 2.3. Integrowanie (a) i łączenie terminologii (b)

Na rysunku 2.3a proces dostosowywania terminologii integrowanych do konkretnych potrzeb został oznaczony strzałkami symbolizującymi transformację terminologii O_{x_1} , O_{x_2} , O_{x_3} i O_{x_4} do terminologii O'_{x_1} , O'_{x_2} , O'_{x_3} i O'_{x_4} , przy czym terminologią wynikową jest O . Terminologie O'_{x_1} , O'_{x_2} , O'_{x_3} i O'_{x_4} mogą opisywać różne lub te same dziedziny. Dziedziny terminologii integrowanych mogą być rozłączne (np. terminologie O'_{x_3} i O'_{x_4}) lub się pokrywać – w pełni lub częściowo (np. terminologie O'_{x_1} i O'_{x_2}). Na rysunku 2.3a dziedzina terminologii wynikowej O również nie jest sumą dziedzin terminologii integrowanych. Dzieje się tak w konsekwencji definiowania dodatkowych terminów spoza dziedzin terminologii integrowanych.

Szczególnym przypadkiem integrowania ontologii jest łączenie ontologii (ang. *ontology merging*) [PINT2001], [PREE2001], [GOCZ2005f] – również w znaczeniu łączenia terminologii. Łączone terminologie nie są dostosowywane do konkretnych potrzeb; terminologia wynikowa nie definiuje żadnych nowych terminów. W rzeczywistości łączenie terminologii jest procesem znajdowania wspólnych terminów pomiędzy różnymi terminologiami i wywiedzenie z nich nowej terminologii wynikowej, która umożliwia współpracę systemów komputerowych opartych na terminologiach łączonych. Proces ten zilustrowano na rysunku 2.3b: terminologie łączone to terminologie O_{x_1} , O_{x_2} , O_{x_3}

i O_{x_4} . Zauważmy, że terminologie łączone w tym procesie nie ulegają zmianie. Strzałki na rysunku 2.3b w przeciwieństwie do rysunku 2.3a oznaczają tylko to, że terminologie łączone stają się częścią terminologii wynikowej O oznaczonej linią przerywaną. Stąd na rysunku 2.3b terminologia wynikowa składa się właśnie z terminologii O_{x_1} , O_{x_2} , O_{x_3} i O_{x_4} , a nie z terminologii O'_{x_1} , O'_{x_2} , O'_{x_3} i O'_{x_4} . Analogicznie jak w przypadku integrowania terminologii, dziedziny terminologii łączonych mogą pozostawać ze sobą w różnych zależnościach. I tak w naszym przykładzie dziedzina terminologii O_{x_3} i O_{x_4} zawiera się w dziedzinie terminologii O_{x_2} .

Terminologia wynikowa „rozumie” wszystkie terminy zdefiniowane w terminologiach łączonych oraz dodatkowo „wie”, jakie zależności między tymi terminami występują. Posłużmy się przykładem.

Przykład 2.3 Łączenie ontologii

Założmy, że łączone są następujące dwie terminologie A i B :

A : $Kobieta \sqcap Mężczyzna \equiv \perp$ B : $\exists maDziecko.\top \sqsubseteq Człowiek$
 $Kobieta \sqcup Mężczyzna \equiv \top$

Ontologia wynikowa powstała z połączenia obu ontologii wygląda następująco:

$Kobieta \sqcap Mężczyzna \equiv \perp$
 $Kobieta \sqcup Mężczyzna \equiv \top$
 $\exists maDziecko.\top \sqsubseteq Człowiek$
 $Kobieta \sqcup Mężczyzna \equiv Człowiek$

Zauważmy, że semantyka terminologii wynikowej nie jest – jak mogłoby się wydawać na pierwszy rzut oka – prostą sumą semantyki terminologii połączonych. Obie terminologie A i B stają się częścią terminologii wynikowej. Dodatkowo, w terminologii wynikowej określona jest również zależność między conceptami *Kobieta* i *Mężczyzna* zdefiniowanymi w terminologii A a conceptem *Człowiek* zdefiniowanym w terminologii B . Zależność ta została zapisana w aksjomacie $Kobieta \sqcup Mężczyzna \equiv Człowiek$ i nie była jawnie wyspecyfikowana w żadnej z terminologii składowych.

Terminologia powstała w wyniku połączenia terminologii może zastąpić terminologie łączone lub być warstwą pośredniczącą pomiędzy systemami opartymi na terminologiach łączonych. Jeśli nowa terminologia pełni rolę warstwy pośredniczącej, mówimy o **odzworowywaniu ontologii** (ang. *ontology mapping*). Analogicznie jak w poprzednich przypadkach, słowo ontologia jest tutaj użyte w znaczeniu terminologii. W przypadku odzworowywania ontologii warstwa pośrednicząca tłumaczy terminy z jednej terminologii na terminy z drugiej terminologii. Proces ten może być manualny, półautomatyczny lub automatyczny [EHRIG2004], [GOCZ2006c]. W przykładzie 2.3 warstwa pośrednicząca musiałaby „wiedzieć”, że dziedziny obu terminologii są takie same, a więc że $Kobieta \sqcup Mężczyzna \equiv Człowiek$.

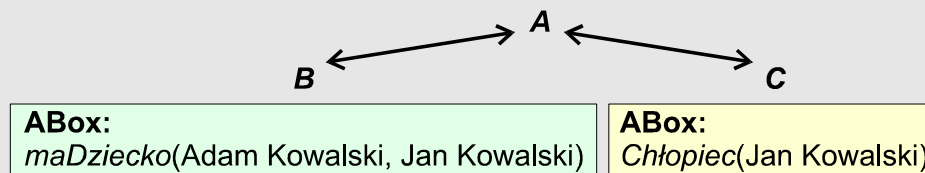
Odnosząc się do problemu braku identyfikowalności zasobów (2), zauważamy, że istniejące systemy wymagają, aby te same zasoby były identyfikowane tym samym URI. Przyjmuje się, że dwa różne URI to dwa różne zasoby. W logice opisowej przyjmuje się założenie **unikatowości nazw** (ang. *unique name assumption* - UNA). Wracając do pojęcia

interpretacji \mathcal{I} , zakładamy, że interpretacja \mathcal{I} nie tylko odwzorowuje koncepty atomowe i role w zbiory i relacje, ale również odwzorowuje każdego osobnika a w element zbioru $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Przy założeniu UNA, jeśli a, b są różnymi nazwami osobników, to $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ [BAAD2003].

Problem ostatni (3), dotyczący łączenia wiedzy pochodzącej z różnych źródeł w Sieci, w sposób zasadniczy odnosi się do zagadnień wnioskowania na zintegrowanych ontologiach (terminologiach i opisach świata). W opisie problemu dotyczącego wielości ontologii (1) zazaczyliśmy, że możliwe jest tworzenie terminologii zintegrowanych występujących fizycznie oraz takich, które są jedynie warstwą pośredniczącą. W pierwszym przypadku, pomijając problem wydajności, proces wnioskowania po ustaleniu jednoznacznych URI dla tych samych zasobów jest problemem analogicznym do problemu wnioskowania dla dowolnych innych ontologii. Niestety, to rozwiązanie, właśnie ze względu na problem wydajności, w Sieci Semantycznej będzie miało prawdopodobnie dużo mniejsze zastosowanie niż integrowanie ontologii poprzez ich odwzorowywanie. Problem wnioskowania przy odwzorowywaniu ontologii został zobrazowany w przykładzie 2.4.

Przykład 2.4 Problem wnioskowania przy odwzorowywaniu ontologii

Żałóżmy, że mamy dwie ontologie B i C o takich samych terminologiach. W opisie świata ontologii B istnieje asercja mówiąca, że *Adam Kowalski* ma dziecko *Jana Kowalskiego*. Niestety, informacja o tym, że *Jan Kowalski* jest chłopcem, jest zapisana tylko w ontologii C . Jeśli ontologia zintegrowana A istniałaby fizycznie, to na podstawie obu tych asercji system wnioskujący bez kłopotu odpowiedziałby na pytanie: „Podaj wszystkich, którzy są rodzicami przynajmniej jednego chłopca” (w konwencji zapisu logiki opisowej: podaj wszystkie wystąpienia konceptu $\exists \text{maDziecko.Chłopiec}$). Jeśli jednak ontologia zintegrowana stanowi jedynie warstwę pośredniczącą pomiędzy ontologiami integrowanymi, to zachodzi pytanie, w jaki sposób system wnioskujący powinien przetworzyć to zapytanie.



Rysunek 2.4. Przykład problemu integracji ontologii poprzez odwzorowywanie ontologii

W następnym podrozdziale opisano, czym jest system integracji ontologii w Sieci Semantycznej i jakie są jego zadania w odniesieniu do zidentyfikowanych powyżej problemów.

2.5 System integracji wiedzy w Sieci Semantycznej

Rozwój Sieci Semantycznej stawia nowe wyzwania w dziedzinie integracji wiedzy. Integracja wiedzy nie jest jednak oderwana od problemu integracji danych, z którego w rzeczywistości się wywodzi. Nie można integrować wiedzy, nie integrując zarazem danych. Pierwsze rozwiązania z dziedziny integracji danych zostały zaproponowane już w latach dziewięćdziesiątych XX wieku [LEVY1998], [LATT1998] i z tych rozwiązań wywodzi się wiele prób rozstrzygnięcia problemu integracji wiedzy.

Celem *systemów integracji danych* jest dostarczenie jednolitego interfejsu do wielu, dostępnych w sieci, heterogenicznych źródeł danych [GRAB2004]. Oznacza to, że systemy

integracji danych mają umożliwiać użytkownikowi zadanie zapytania, uwalniając go od poszukiwania odpowiedzi. Stąd wyszukiwanie odpowiednich źródeł danych, w których może być zawarta częściowa lub pełna odpowiedź na zadane pytanie, interakcja z każdym z tych źródeł z osobna i scalanie uzyskanych informacji w spójną odpowiedź są zadaniami systemu integracji danych, nie zaś jego użytkownika.

Problem integracji danych jest zagadnieniem złożonym, co spowodowane jest:

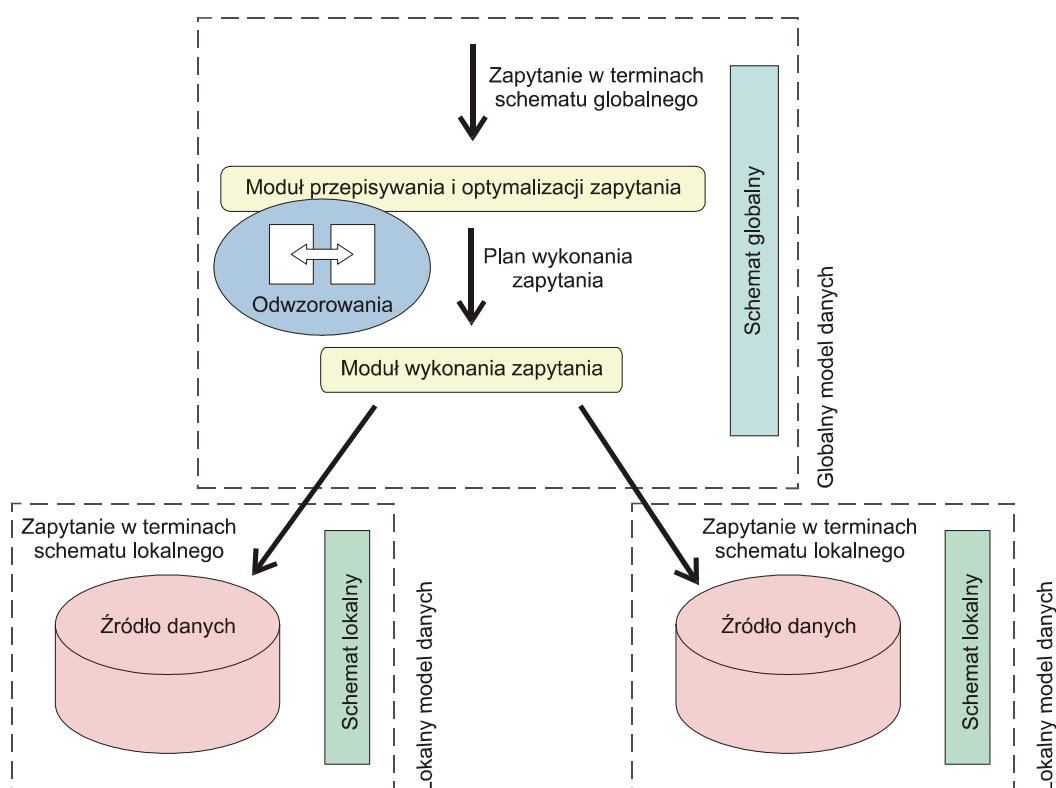
- 1) powiązaniem i nakładaniem się danych zapisanych w różnych źródłach,
 - 2) niejednorodnymi modelami i schematami danych,
 - 3) różnorodnymi możliwościami przetwarzania zapytań przez źródła danych
- [LEVY2000].

Rozwiązanie problemu integracji danych wymaga zdefiniowania i opracowania dwóch zagadnień:

- 1) sposobów opisywania zawartości różnorodnych źródeł danych,
- 2) procesu uzyskiwania odpowiedzi na zapytania kierowane do systemu integracji danych, na podstawie wcześniej zdefiniowanego opisu źródeł [BAAD2003].

Aktualnie wydaje się, że serwery relacyjnych baz danych dostarczają najbardziej efektywnego sposobu przetwarzania danych rozumianego jako dodawanie, aktualizacja i odczytywanie danych. Są one odpowiednią podstawą do budowania systemów integracji danych. Nie dostarczają jednak możliwości semantycznego opisywania danych oraz wnioskowania z tych danych, a tym samym nie są odpowiednie do budowania systemów integracji wiedzy.

W 2000 roku Levy zaproponował architekturę systemu integracji danych [LEVY2000], która w rzeczywistości jest architekturą odpowiednią również dla systemów integracji wiedzy. Ta architektura została przedstawiona na rysunku 2.5.

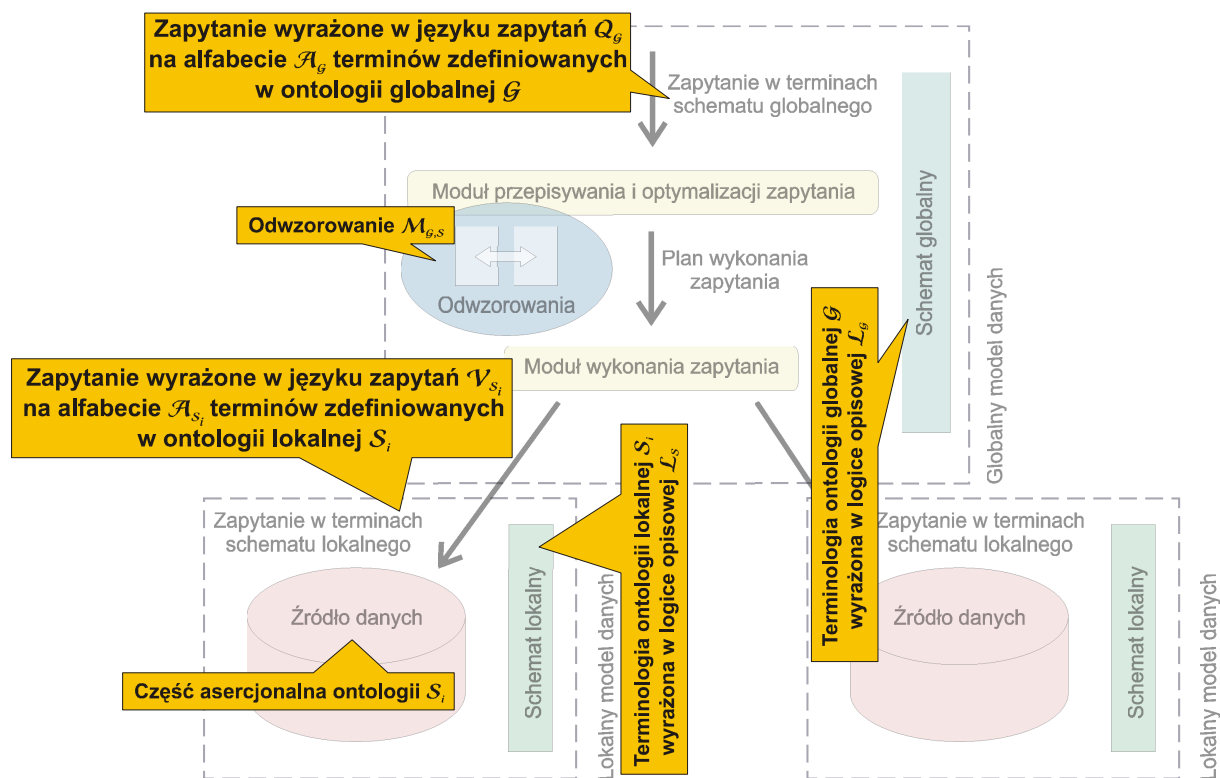


Rysunek 2.5. Architektura systemu integracji danych

System integracji danych bazuje na dwóch modelach danych: modelu globalnym i modelu lokalnym. Zapytanie zawsze zadawane jest w terminach schematu globalnego. Moduł przepisywania i optymalizacji zapytania jest odpowiedzialny za przetłumaczenie zapytania zadanego w terminach schematu globalnego na zapytania zadawane w terminach schematów lokalnych. Również zapytania te muszą być zoptymalizowane pod kątem wydajności. Aby możliwe było przepisanie zapytania zadanego w terminach schematu globalnego do zbioru zapytań zadanego w terminach schematów lokalnych, musi istnieć odwzorowanie pomiędzy terminami globalnymi i terminami lokalnymi. System integracji danych zaproponowany przez Levy'ego bazuje na modelu relacyjnym, zarówno dla modelu globalnego, jak i modeli lokalnych, stąd odwzorowania są definiowane również w terminach relacji bazodanowych. Przedstawione są dwa główne podejścia do budowania takich odwzorowań. Pierwsze z nich to podejście zwane **Global As View** (GAV) [CHAWA1994], w którym każdej relacji w schemacie globalnym przyporządkowujemy zapytania na relacjach lokalnych określające, w jaki sposób uzyskać krotki relacji ze źródeł lokalnych. Drugie z nich to podejście **Local As View** (LAV) [LEVY2001], w którym opisy źródeł są definiowane w przeciwnym kierunku: relacjom lokalnym przypisywane jest zapytanie w terminach relacji globalnych. Każde z tych podejść ma swoje wady i zalety:

- 1) w podejściu GAV przekształcenie zapytania z terminów globalnych na terminy lokalne jest bardzo proste, jednakże dodanie nowego źródła lokalnego wymaga przejrzenia wszystkich odwzorowań i rozszerzenia ich o nowe źródło lokalne;
- 2) w podejściu LAV przekształcenie zapytania z terminów globalnych na terminy lokalne jest procesem złożonym, często wymagającym skomplikowanych algorytmów, jednakże dodanie nowego źródła jest niezależne od istnienia innych źródeł i nie wymaga przejrzenia wcześniej utworzonych odwzorowań.

Odpowiednio przygotowany plan wykonania zapytania jest przesyłany do modułu wykonania zapytania.



Rysunek 2.6. Wykorzystanie architektury systemu integracji danych w systemie integracji wiedzy

W 2001 roku Calvanese [CALV2001] zaproponował nową architekturę systemu integracji wiedzy. Calvanese nazywa ten system systemem integracji ontologii (ang. *Ontology Integration System* - OIS), w którym ontologia globalna nadrzędna (integrująca) umożliwia opisanie wiedzy zawartej w różnych ontologiach lokalnych (integrowanych). Architektura przedstawiona przez Levy'ego jest systemem integracji danych (schematy lokalne i schematy globalne to schematy relacyjne), zaś architektura przedstawiona przez Calvanese to niejako ta sama architektura, ale dostosowana dla schematu globalnego i schematów lokalnych będących terminologiami. Architektura systemu integracji ontologii będąca rozszerzeniem architektury systemu integracji danych została przedstawiona na rysunku 2.6.

Poniżej przytoczono formalną definicję systemu integracji ontologii zaproponowanego przez Calvanese oraz opisano zależności pomiędzy poszczególnymi komponentami systemu integracji ontologii i systemu integracji danych.

Definicja 2.6 System integracji ontologii OIS

System integracji ontologii OIS jest trójką postaci $\langle \mathcal{G}, \mathcal{S}, \mathcal{M}_{\mathcal{G},\mathcal{S}} \rangle$, w której \mathcal{G} jest ontologią globalną, \mathcal{S} zbiorem ontologii lokalnych oraz $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ jest odwzorowaniem pomiędzy \mathcal{G} i ontologiami lokalnymi z \mathcal{S} .

W odniesieniu do architektury Levy'ego terminologia ontologii globalnej jest schematem globalnym, zaś terminologie ontologii lokalnych to schematy lokalne poszczególnych źródeł. Odwzorowanie $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ to również nic innego jak zbiór odwzorowań pomiędzy terminami globalnymi i lokalnymi. Calvanese definiuje formalnie ontologie lokalne i ontologię globalną w następujący sposób:

Definicja 2.7 Ontologia lokalna \mathcal{S}_i

Ontologia lokalna \mathcal{S}_i jest dowolną ontologią zdefiniowaną w zbiorze ontologii lokalnych \mathcal{S} wyrażoną w logice opisowej $\mathcal{L}_{\mathcal{S}_i}$ na alfabecie $\mathcal{A}_{\mathcal{S}_i}$.

Definicja 2.8 Ontologia globalna \mathcal{G}

Ontologia globalna \mathcal{G} jest ontologią wyrażoną w logice opisowej $\mathcal{L}_{\mathcal{G}}$ na alfabecie $\mathcal{A}_{\mathcal{G}}$.

Dodatkowo Calvanese nakłada warunek na alfabety ontologii lokalnych i ontologii globalnej. Różne $\mathcal{A}_{\mathcal{S}_i}$ są wzajemnie rozłączne i każdy z alfabetów $\mathcal{A}_{\mathcal{S}_i}$ jest rozłączny z alfabetem $\mathcal{A}_{\mathcal{G}}$.

Calvanese definiuje ontologię globalną jako zunifikowany widok umożliwiający formułowanie zapytań w terminach ontologii lokalnych. Stąd, choć nie jest to jednoznacznie ujęte w modelu systemu OIS, przyjmujemy, że nie istnieją asercje wypowiedziane w terminach tej ontologii globalnej. Dodatkowo, każdy alfabet jest zależny od logiki opisowej, w której dana ontologia została zdefiniowana. Przykładowo, kiedy mówimy o alfabecie dla języka \mathcal{ALC} , w przypadku ontologii globalnej mówimy o zbiorze terminów będących nazwami konceptów i ról atomowych, zaś w przypadku ontologii lokalnych również nazw osobników. Na rysunku 2.6 schemat globalny został nazwany terminologią ontologii globalnej w celu uniknięcia wieloznaczności, jednakże ontologia globalna rozumiana jako widok nie zawiera asercji i tym samym jest wyłącznie terminologią.

Sercem systemu OIS jest odwzorowanie $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ definiujące sposób wzajemnego odwzorowania pomiędzy konceptami w ontologii globalnej \mathcal{G} a konceptami w ontologiach lokalnych \mathcal{S} . W rzeczywistości jednak definicja ta jest pewnym uproszczeniem i już dla języka \mathcal{ALC} wymaga rozszerzenia o role. Ogólnie możemy przyjąć, że odwzorowanie $\mathcal{M}_{\mathcal{G},\mathcal{S}}$

definiuje sposób wzajemnego odwzorowania relacji \mathcal{R} w ontologii globalnej \mathcal{G} i w ontologiach lokalnych \mathcal{S}_i . Przez relacje unarne rozumiemy koncepty, przez relacje binarne role, a dla bardziej ekspresywnych logik opisowych również relacje n -arne.

Posługując się pojęciami logiki opisowej, Calvanese definiuje semantykę systemu OIS. Rozważmy model lokalny \mathcal{D} dla $\mathcal{O} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M}_{\mathcal{G},\mathcal{S}} \rangle$, tj. taką interpretację, która jest modelem dla każdej ontologii z \mathcal{S} . Przez interpretację globalną dla \mathcal{O} rozumiemy dowolną interpretację dla \mathcal{G} .

Definicja 2.9 Model globalny \mathcal{O} względem \mathcal{D}

Interpretacja globalna \mathcal{I} dla \mathcal{O} jest modelem globalnym dla \mathcal{O} względem \mathcal{D} , jeśli:

- 1) \mathcal{I} jest modelem \mathcal{G} ;
 - 2) \mathcal{I} spełnia odwzorowanie $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ względem \mathcal{D} .
-

W taki sposób możemy zdefiniować semantykę systemu OIS:

Definicja 2.10 Semantyka systemu OIS

Semantykę systemu OIS $\text{sem}(\mathcal{O})$ definiujemy jako:

$$\text{sem}(\mathcal{O}) = \{ \mathcal{I} : \text{istnieje model lokalny } \mathcal{D} \text{ dla } \mathcal{O} \text{ taki, że } \mathcal{I} \text{ jest modelem globalnym dla } \mathcal{O} \text{ względem } \mathcal{D} \}$$

Pozostaje zdefiniowanie pojęcie spełnialności odwzorowania $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ przez \mathcal{I} względem \mathcal{D} . Definicja ta jest jednak zależna od zastosowanego podejścia do budowania odwzorowań. Tak jak w przypadku systemów integracji danych, tak i dla systemu integracji ontologii mówimy o budowaniu odwzorowania zgodnie z podejściem GAV albo LAV. W podejściu GAV relacje z ontologii globalnej \mathcal{G} są odwzorowywane w zapytania do ontologii lokalnych z \mathcal{S} . W podejściu LAV relacje z ontologii lokalnych z \mathcal{S} są odwzorowywane w zapytania do ontologii globalnej \mathcal{G} .

W podejściu GAV zakładamy, że mamy język zapytań $\mathcal{V}_{\mathcal{S}}$ na alfabecie $\mathcal{A}_{\mathcal{S}}$. Każdej nazwanej relacji \mathcal{R} w \mathcal{G} przyporządkowujemy pewne zapytanie $V_{\mathcal{S}}$ do \mathcal{S} takie, że odwzorowanie pomiędzy \mathcal{R} i $V_{\mathcal{S}}$ jest *poprawne* (ang. *sound*) i *kompletne* (ang. *complete*).

Definicja 2.11 Spełnialność odwzorowania $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ przez \mathcal{I} względem \mathcal{D} w podejściu GAV

Niech \mathcal{D} będzie modelem lokalnym \mathcal{O} , a \mathcal{I} globalną interpretacją \mathcal{O} .

- 1) \mathcal{I} spełnia trójkę $\langle \mathcal{R}, V_{\mathcal{S}}, \text{poprawne} \rangle$ w $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ względem \mathcal{D} , jeśli wszystkie krotki spełniające $V_{\mathcal{S}}$ w \mathcal{D} spełniają \mathcal{R} w \mathcal{I} ;
- 2) \mathcal{I} spełnia trójkę $\langle \mathcal{R}, V_{\mathcal{S}}, \text{kompletne} \rangle$ w $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ względem \mathcal{D} , jeśli nie jest prawdą, że istnieją krotki niespełniające $V_{\mathcal{S}}$ w \mathcal{D} , a spełniające \mathcal{R} w \mathcal{I} .

Mówimy, że \mathcal{I} spełnia odwzorowanie $\mathcal{M}_{\mathcal{G},\mathcal{S}}$ względem \mathcal{D} , jeśli spełnia trójkę $\langle \mathcal{R}, V_{\mathcal{S}}, \text{poprawne} \rangle$ oraz trójkę $\langle \mathcal{R}, V_{\mathcal{S}}, \text{kompletne} \rangle$.

W podejściu LAV zakładamy, że mamy język zapytań $\mathcal{V}_{\mathcal{G}}$ na alfabecie $\mathcal{A}_{\mathcal{G}}$. Każdej nazwanej relacji \mathcal{R} w \mathcal{S} przyporządkowujemy pewne zapytanie $V_{\mathcal{G}}$ do \mathcal{G} takie, że odwzorowanie pomiędzy \mathcal{R} i $V_{\mathcal{G}}$ jest poprawne i kompletne.

Definicja 2.12 Spełnialność odwzorowania $M_{G,S}$ przez \mathcal{I} względem \mathcal{D} w podejściu LAV

Niech \mathcal{D} będzie modelem lokalnym \mathcal{O} , a \mathcal{I} globalną interpretacją \mathcal{O} .

- 1) \mathcal{I} spełnia trójkę $\langle V_g, \mathcal{R}, \text{poprawne} \rangle$ w $M_{G,S}$ względem \mathcal{D} , jeśli wszystkie krotki spełniające \mathcal{R} w \mathcal{D} spełniają V_g w \mathcal{I} ;
- 2) \mathcal{I} spełnia trójkę $\langle V_g, \mathcal{R}, \text{kompletne} \rangle$ w $M_{G,S}$ względem \mathcal{D} , jeśli nie jest prawdą, że istnieją krotki niespełniające \mathcal{R} w \mathcal{D} , a spełniające V_g w \mathcal{I} .

Mówimy, że \mathcal{I} spełnia odwzorowanie $M_{G,S}$ względem \mathcal{D} , jeśli spełnia trójkę $\langle V_g, \mathcal{R}, \text{poprawne} \rangle$ oraz trójkę $\langle V_g, \mathcal{R}, \text{kompletne} \rangle$.

Niestety ekspresywność obu tych podejść jest ograniczona. Dlatego poza rozwiązaniami opartymi na GAV i LAV zaproponowano również tworzenie odwzorowań nieograniczonych. Tworzenie tych odwzorowań polega na odwzorowywaniu zapytań V_g do ontologii globalnej w zapytania V_s do ontologii lokalnych. Analogicznie jak w przypadku podejść GAV i LAV, takie odwzorowanie pomiędzy V_g i V_s musi być poprawne i kompletne.

Definicja 2.13 Spełnialność odwzorowania $M_{G,S}$ przez \mathcal{I} względem \mathcal{D} w podejściu nieograniczonym

Niech \mathcal{D} będzie modelem lokalnym \mathcal{O} , a \mathcal{I} globalną interpretacją \mathcal{O} .

- 1) \mathcal{I} spełnia trójkę $\langle V_g, V_s, \text{poprawne} \rangle$ w $M_{G,S}$ względem \mathcal{D} , jeśli wszystkie krotki spełniające V_s w \mathcal{D} spełniają V_g w \mathcal{I} ;
- 2) \mathcal{I} spełnia trójkę $\langle V_g, V_s, \text{kompletne} \rangle$ w $M_{G,S}$ względem \mathcal{D} , jeśli nie jest prawdą, że istnieją krotki niespełniające V_s w \mathcal{D} , a spełniające V_g w \mathcal{I} .

Mówimy, że \mathcal{I} spełnia odwzorowanie $M_{G,S}$ względem \mathcal{D} , jeśli spełnia trójkę $\langle V_g, V_s, \text{poprawne} \rangle$ oraz trójkę $\langle V_g, V_s, \text{kompletne} \rangle$.

Niestety podejście to wykazuje wady zarówno podejścia GAV, jak i LAV.

Aby odnieść architekturę systemu integracji ontologii do systemu integracji danych, pozostaje jeszcze określić, czym w systemie OIS jest zapytanie. Zapytania wyrażane są w języku zapytań Q_G na alfabecie \mathcal{A}_G i mają na celu wydobywanie zbioru krotek złożonych z elementów Δ . Dlatego każde zapytanie ma określoną krotność, a znaczenie zapytania q o krotności n jest zdefiniowane następująco:

Definicja 2.14 Znaczenie zapytania w systemie OIS

Odpowiedź $q^{\mathcal{O}}$ na zapytanie q dla \mathcal{O} jest zbiorem krotek:

$$q^{\mathcal{O}} = \{ \langle c_1, \dots, c_n \rangle : \text{dla każdego } \mathcal{I} \in \text{sem}(\mathcal{O}), \\ \langle c_1, \dots, c_n \rangle \in q^{\mathcal{I}} \},$$

gdzie $q^{\mathcal{I}}$ oznacza wynik zapytania q w interpretacji \mathcal{I} .

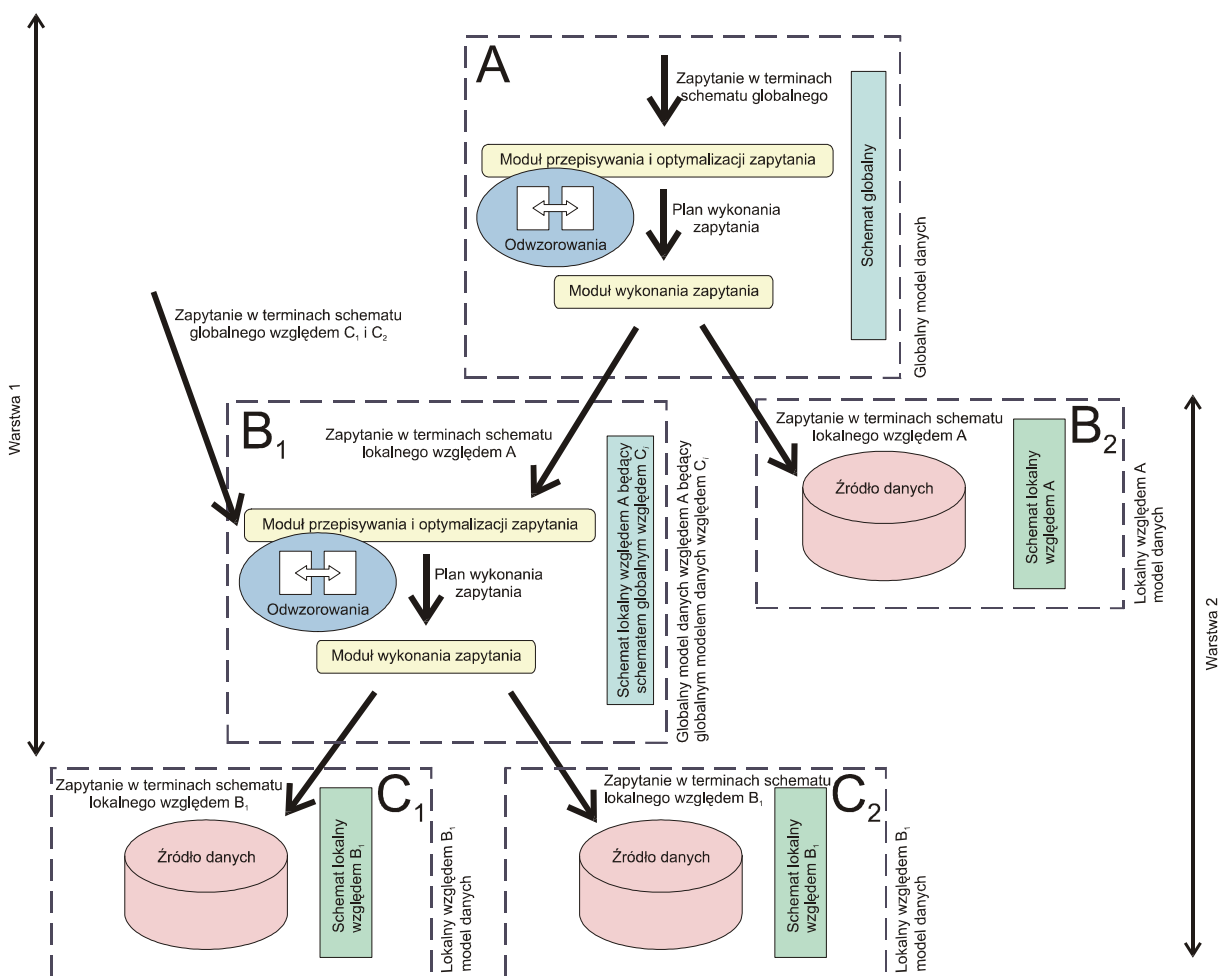
Przedstawiony model systemu integracji jest modelem jednowarstwowym. Możliwe jest rozszerzenie tego modelu do systemu wielowarstwowego (co nie zostało ujęte w modelu Calvanese), w którym pewne ontologie lokalne mogą stać się ontologiami nadrzędnymi (globalnymi) dla innych ontologii. Przykład systemu wielowarstwowego został przedstawiony na rysunku 2.7. W systemie wielowarstwowym mówimy o schemacie globalnym i schematach lokalnych dla poszczególnych warstw. Warstwa 1 składa się z trzech składowych: A , B_1 , B_2 . W tej warstwie B_1 i B_2 reprezentują składowe lokalne: zapytania są zadawane w terminach schematów lokalnych względem A . W warstwie 2 możemy wyróżnić jeden jednowarstwowy system integracji ontologii składający się ze składowych: B_1 , C_1 i C_2 .

Składowa B_1 jest składową lokalną dla warstwy 1 i składową globalną dla warstwy 2. W warstwie 2 znajduje się również składowa B_2 , która w warstwie 2 nie jest elementem żadnego podsystemu integracji ontologii (jednowarstwowego systemu integracji ontologii należącego do pewnej warstwy). Zastosowanie wielowarstwowego systemu integracji danych ma liczne zalety, zwłaszcza w zastosowaniach praktycznych, takie jak:

- 1) rozproszenie procesu wnioskowania,
- 2) możliwość zadawania zapytań skierowanych do odpowiednich warstw,
- 3) brak konieczności przechowywania danych w jednym miejscu.

Zaproponowany przez Calvanese model systemu integracji stanowi podstawę modelu systemu integracji wiedzy zaproponowanego w rozdziale 3. Model zaproponowany w rozprawie uściśla model Clavanese przez:

- 1) zdefiniowanie sposobu budowania odwzorowań,
- 2) wybranie odpowiedniego języka opisu,
- 3) wykorzystanie odpowiednich algorytmów wnioskujących,
- 4) wykorzystanie odpowiedniego języka zapytań.



Rysunek 2.7. Wielowarstwowa architektura systemu integracji ontologii

2.6 Podsumowanie

Teza rozprawy w sposób szczególny odnosi się do praktycznego zastosowania zaproponowanych w rozprawie metod integracji ontologii. W przedstawionym rozdziale, poprzez naszkicowanie charakteru Sieci Semantycznej, określono wymagania względem systemu integracji wiedzy, wynikające z jego umiejscowienia w środowisku Sieci Semantycznej.

Aktualnie, głównym językiem opisu ontologii w Sieci Semantycznej jest język OWL, a w szczególności dialekt tego języka oparty na logice opisowej. Dlatego że logika opisowa jest podstawową metodą reprezentacji wiedzy w Sieci Semantycznej w niniejszym rozdziale poświęcono jej wiele uwagi.

Ostatnia część rozdziału zawiera opis istniejącego, zaproponowanego przez Calvanese, modelu systemu integracji ontologii. Jednym z celów rozprawy jest przedstawienie modelu systemu integracji wiedzy. Model Calvanese jest punktem odniesienia dla przedstawionego w rozprawie modelu i dlatego w rozdziale zawarto formalny opis zaproponowanego przez Calvanese modelu.

W kolejnym rozdziale przedstawiono nowy model systemu integracji wiedzy, stanowiący podstawę do wyników zaprezentowanych w następnych rozdziałach rozprawy.

Rozdział 3. System integracji wiedzy

3.1 Model systemu integracji wiedzy

Przez system integracji wiedzy rozumiemy system integrujący (czyli scalający w sposób logicznie spójny) wiedzę z różnych źródeł wiedzy. We wstępie do niniejszej rozprawy źródło wiedzy zostało określone jako zbiór danych lub informacji podanych w formie zrozumiałej dla odbiorcy (człowieka lub komputera). Według nauki o zarządzaniu wiedzą źródło wiedzy możemy postrzegać jako system dostarczający informacje w terminach pewnej ontologii. Taki system jest źródłem wiedzy dla innych systemów „rozumiejących” tę ontologię i znających metodę komunikacji z tym systemem. Taką metodę komunikacji nazwijmy *językiem zapytań*. W dalszej części tego podrozdziału wprowadzimy niezbędne formalne definicje takich pojęć, jak ontologia, język zapytań i źródło wiedzy.

Definicja 3.1 Ontologia O

Ontologią O wyrażoną w pewnej logice opisowej \mathcal{L}_O nazywamy parę $\langle O_x, O_s \rangle$. Niech \mathcal{A}_O oznacza alfabet terminów zdefiniowanych w ontologii O . $\mathcal{A}_O = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A} \cup \mathcal{A}_{O_I}$ jest zbiorem terminów zdefiniowanych w czterech alfabetach: alfabecie \mathcal{A}_{O_C} terminów będących nazwami konceptów, alfabecie \mathcal{A}_{O_R} terminów będących nazwami ról, alfabecie \mathcal{A}_{O_A} terminów będących nazwami atrybutów i alfabecie \mathcal{A}_{O_I} terminów będących nazwami osobników. Wówczas O_x (TBox, terminologia ontologii O) jest zbiorem aksjomatów wyrażonych w terminach alfabetu $\mathcal{A}_{O_T} = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A}$, zaś O_s (ABox, asercje dotyczące osobników ontologii O) jest zbiorem asercji wyrażonych w terminach alfabetu \mathcal{A}_O .

Przykład 3.1 Ontologia Pracowników

Określimy przykładową ontologię opisującą pracowników. Terminologia O_x jest zdefiniowana następująco:

$$Kobieta \sqcup Mężczyzna \equiv Człowiek \quad (1)$$

$$Kobieta \sqcap Mężczyzna \equiv \perp \quad (2)$$

$$Pracownik \sqsubseteq Człowiek \quad (3)$$

$$\text{Pracownik} \sqsubseteq \exists \text{jestNaStanowisku}.\text{Stanowisko} \quad (4)$$

$$\exists \text{jestNaStanowisku}.\top \sqsubseteq \text{Pracownik} \quad (5)$$

$$\exists \text{jestNaStanowisku}.\neg \text{Stanowisko} \sqsubseteq \perp \quad (6)$$

$$\text{Pracownik} \sqsubseteq \exists \text{dostajeWynagrodzenie}.\langle 1000, 8000 \rangle \quad (7)$$

$$\exists \text{dostajeWynagrodzenie}.(0, \infty) \sqsubseteq \text{Pracownik} \quad (8)$$

$$\text{Stanowisko} \sqcap \text{Człowiek} \equiv \perp$$

W terminologii zdefiniowane są koncepty atomowe: *Człowiek*, *Kobieta*, *Mężczyzna*, *Pracownik* i *Stanowisko*. Człowiek może być albo kobietą, albo mężczyzną (1), (2). Wśród ludzi są pracownicy (3). Każdy pracownik jest na pewnym stanowisku (4). *jestNaStanowisku* jest rolą, której dziedziną jest koncept *Pracownik* (5), a zakresem koncept *Stanowisko* (6). Dodatkowo, każdy pracownik dostaje wynagrodzenie od 1000 do 8000 złotych (7), co jest wyrażone przy użyciu atrybutu *dostajeWynagrodzenie*. Dziedziną atrybutu *dostajeWynagrodzenie* jest również koncept *Pracownik*.

Dla tak zdefiniowanej terminologii stwórzmy przykładowy zbiór asercji:

$$\text{Kobieta}(\text{AnnaKowalska}) \quad (1)$$

$$\text{Mężczyzna}(\text{JanKowalski}) \quad (2)$$

$$\text{Stanowisko}(\text{Prezes}) \quad (3)$$

$$\text{dostajeWynagrodzenie}(\text{JanKowalski}, 2000) \quad (4)$$

$$\text{jestNaStanowisku}(\text{JanKowalski}, \text{Prezes}) \quad (5)$$

W języku naturalnym możemy wyrazić powyższe asercje jako zbiór następujących zdań:

$$\text{Anna Kowalska jest kobietą.} \quad (1)$$

$$\text{Jan Kowalski jest mężczyzną.} \quad (2)$$

$$\text{Prezes jest nazwą stanowiska.} \quad (3)$$

$$\text{Jan Kowalski dostaje wynagrodzenie wysokości 2000.} \quad (4)$$

$$\text{Jan Kowalski zajmuje stanowisko prezesa} \quad (5)$$

Alfabety \mathcal{A}_{O_C} , \mathcal{A}_{O_R} , \mathcal{A}_{O_A} i \mathcal{A}_{O_I} są następującymi zbiorami:

$$\mathcal{A}_{O_C} = \{\text{Kobieta}, \text{Mężczyzna}, \text{Człowiek}, \text{Pracownik}, \text{Stanowisko}\},$$

$$\mathcal{A}_{O_R} = \{\text{jestNaStanowisku}\},$$

$$\mathcal{A}_{O_A} = \{\text{dostajeWynagrodzenie}\}.$$

$$\mathcal{A}_{O_I} = \{\text{AnnaKowalska}, \text{JanKowalski}, \text{Prezes}\}.$$

Zatem ontologia O jest wyrażona alfabetem $\mathcal{A}_O = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A} \cup \mathcal{A}_{O_I}$:

$$\mathcal{A}_O = \{\text{Kobieta}, \text{Mężczyzna}, \text{Człowiek}, \text{Pracownik}, \text{Stanowisko}, \text{jestNaStanowisku}, \\ \text{dostajeWynagrodzenie}, \text{AnnaKowalska}, \text{JanKowalski}, \text{Prezes}\},$$

zaś terminologia O_x alfabetem $\mathcal{A}_{O_T} = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A}$:

$$\mathcal{A}_{O_T} = \{\text{Kobieta}, \text{Mężczyzna}, \text{Człowiek}, \text{Pracownik}, \text{Stanowisko}, \text{jestNaStanowisku}, \\ \text{dostajeWynagrodzenie}\}.$$

Definicja 3.2 Język zapytań Q_O

Językiem zapytań $Q_O = Q_{O_T} \cup Q_{O_I}$ na alfabecie \mathcal{A}_O nazywamy zbiór zapytań składający się ze zbioru zapytań terminologicznych Q_{O_T} i zbioru zapytań asercjonalnych Q_{O_I} .

Źródła wiedzy mogą się między sobą różnić zbiorem obsługiwanych zapytań. W systemie integracji wiedzy przyjmujemy, że wszystkie źródła wiedzy integrowane w ramach opisywanego systemu „rozumieją” pewien ustalony język zapytań Q_O .

Definicja 3.3 Źródło wiedzy

Źródłem wiedzy $\mathcal{KS} = \langle O, Q_O \rangle$ nazywamy system opisujący wiedzę w terminach ontologii O i dostarczający metod wnioskowania umożliwiających poprawne i kompletne odpowiadanie na zapytania sformułowane w języku zapytań Q_O .

Mówimy tu o poprawności (ang. *soundness*) i kompletności (ang. *completeness*) odpowiedzi. Przykładowo, dla zapytań asercjonalnych odpowiedź jest poprawna, jeśli każdy osobnik, którego nazwa jest zwrócona, spełnia warunki postawione w zapytaniu. Odpowiedź jest kompletna, jeśli nie istnieje osobnik, którego nazwa nie została zwrócona w zapytaniu, a który spełnia warunki w tym zapytaniu postawione.

W systemie integracji wiedza pochodzi z wielu źródeł wiedzy. Zdefiniujemy pewien zbiór źródeł wiedzy $S_{\mathcal{KS}} = \{\mathcal{KS}_1, \dots, \mathcal{KS}_n\}$, z których ta wiedza pochodzi. Poniżej zebrano ograniczenia, jakim podlegają źródła wiedzy $\mathcal{KS}_1, \dots, \mathcal{KS}_n$:

- 1) Źródła wiedzy „rozumieją” pewien ustalony język zapytań Q_O .
- 2) Dowolne źródło wiedzy z $S_{\mathcal{KS}}$ opisuje wiedzę w terminach pewnej terminologii O_x ze zbioru terminologii $S_O = \{O_{x_1}, \dots, O_{x_m}\}$.
- 3) Każda terminologia ze zbioru S_O jest wyrażona w tej samej logice opisowej \mathcal{L}_O .
- 4) Wiedza z dowolnej liczby źródeł może być opisana tą samą terminologią $O_x \in S_O$.
- 5) Dla terminologii ze zbioru S_O jest zdefiniowany sposób ich łączenia.

Łączenie terminologii ze zbioru S_O rozumiemy w ten sposób, że:

- 1) dla dowolnych dwóch konceptów z terminologii ze zbioru S_O jest określona zależność, jaka zachodzi między nimi (rozłączność, tożsamość lub posiadanie określonej części wspólnej),
- 2) dla dowolnych dwóch ról z terminologii ze zbioru S_O jest określona zależność, jaka zachodzi między nimi (rozłączność, tożsamość lub posiadanie określonej części wspólnej),
- 3) dla dowolnych dwóch atrybutów z terminologii ze zbioru S_O jest określone, czy są one tożsame.

Sposób łączenia między terminologiami jest opisany w *zbiorze odwzorowań aksjomatycznych*.

Definicja 3.4 Zbiór odwzorowań aksjomatycznych \mathcal{Y}

Zbiorem odwzorowań aksjomatycznych \mathcal{Y} pomiędzy terminologią O_{x_i} i O_{x_j} nazywamy terminologię \mathcal{T} składającą się ze zbioru aksjomatów $Aksjomat(A_1, \dots, A_n)$ wyrażonych w logice opisowej \mathcal{L}_O w terminach A_1, \dots, A_n alfabetu $\mathcal{A}_{O_{ij_T}} = \mathcal{A}_{O_{iT}} \cup \mathcal{A}_{O_{jT}}$ spełniających warunek:

$$\forall Aksjomat(A_1, \dots, A_n) \subseteq \mathcal{T} (\exists A_p \in \mathcal{A}_{O_{iT}} \wedge \exists A_q \in \mathcal{A}_{O_{jT}} \wedge p, q \in \{1, \dots, n\}).$$

Zbiór odwzorowań aksjomatycznych \mathcal{Y} jednoznacznie określa zależności między konceptami, rolami i atrybutami z dwóch łączonych terminologii. W następnym kroku zdefiniujemy znaczenie trójki $(O_{x_i}, O_{x_j}, \mathcal{Y})$.

Definicja 3.5 Znaczenie trójki $(O_{x_i}, O_{x_j}, \mathcal{Y})$

Interpretacja \mathcal{I} jest modelem trójki $(O_{x_i}, O_{x_j}, \mathcal{Y})$, jeżeli jest modelem terminologii O_{x_i} , terminologii O_{x_j} i terminologii \mathcal{T} dla \mathcal{Y} .

Odnieśmy znaczenie trójki $(O_{x_i}, O_{x_j}, \mathcal{Y})$ do znaczenia importu ontologii zdefiniowanego w języku OWL [OWL2004]. Niech w terminologii O_{x_1} jest zdefiniowany pewien zbiór aksjomatów $Z_1 = Z(O_{x_1})$, zaś w terminologii O_{x_2} jest zdefiniowany pewien zbiór aksjomatów $Z_2 = Z(O_{x_2})$. Jeżeli terminologia O_{x_1} nie importuje ontologii O_{x_2} , to modelem terminologii O_{x_1} jest każda interpretacja \mathcal{I} spełniająca zbiór aksjomatów Z_1 . Jeżeli ontologia O_{x_1} importuje terminologię O_{x_2} , to modelem terminologii O_{x_1} jest każda interpretacja \mathcal{I} spełniająca zbiór aksjomatów $Z_1 \cup Z_2$. Niech terminologia O definiuje pusty zbiór aksjomatów Z i importuje terminologie $O_{x_i}, O_{x_j}, \mathcal{T}$. Znaczenie terminologii O jest identyczne jak znaczenie trójki $(O_{x_i}, O_{x_j}, \mathcal{Y})$. W ten sposób pokazaliśmy, w jaki sposób problem łączenia terminologii, przy założeniu definicji zbioru odwzorowań aksjomatycznych 3.4, został sprowadzony do problemu zarządzania ontologiami powiązanymi między sobą relacją importu.

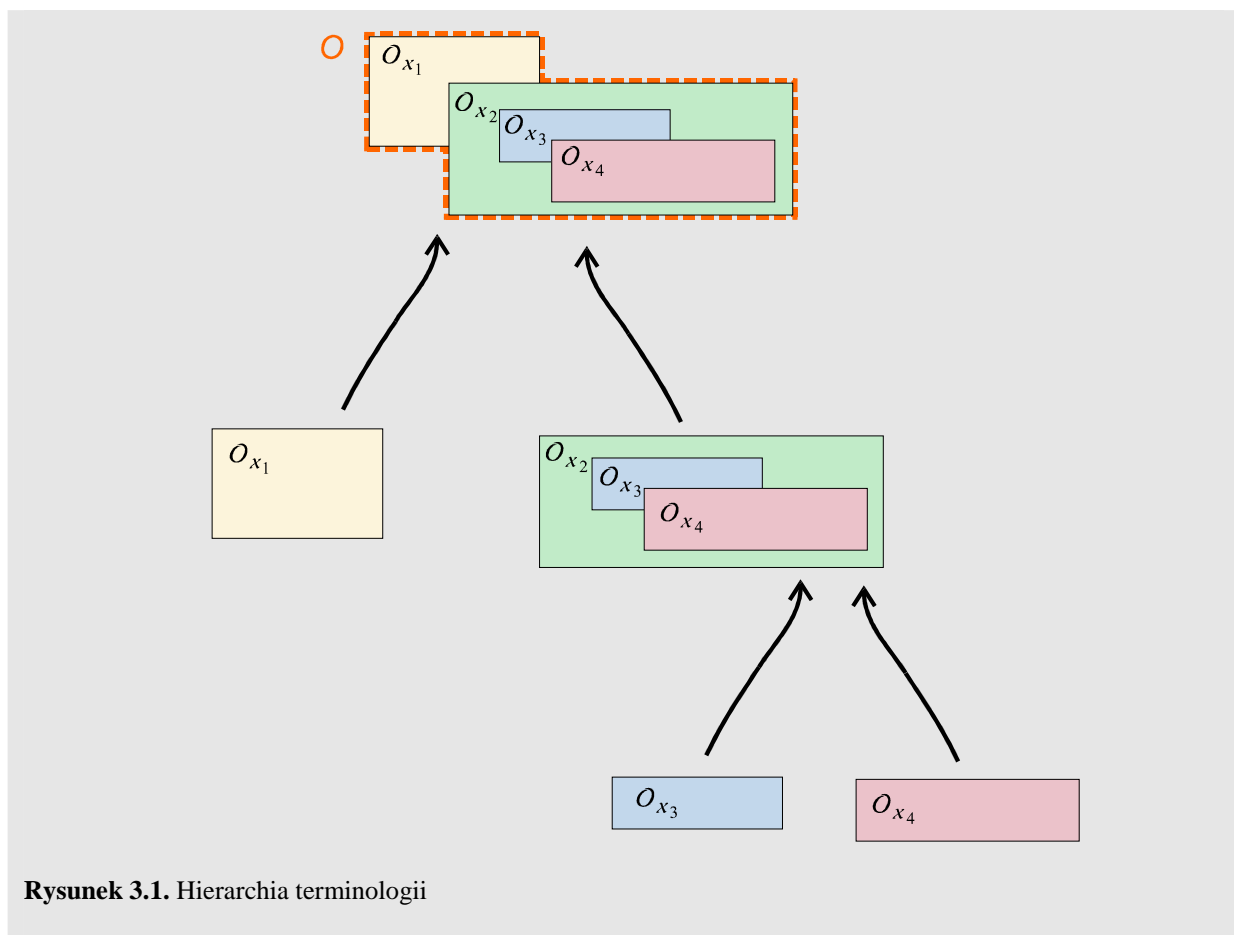
System integracji wiedzy wymaga zbudowania hierarchii terminologii, którymi jest opisana wiedza zapisana w źródłach wiedzy. W celu zdefiniowania pojęcia hierarchii terminologii posłużymy się przykładem.

Przykład 3.2 Budowanie hierarchii terminologii w systemie integracji wiedzy

Na rysunku 2.3.b pokazano łączenie przykładowych terminologii O_{x_1}, \dots, O_{x_4} . Przyjmijmy, że $S_O = \{O_{x_1}, \dots, O_{x_4}, O\}$, gdzie:

- O_{x_2} importuje terminologie O_{x_3} i O_{x_4} ,
- O importuje terminologie O_{x_1} i O_{x_2} .

Analiza zależności między terminologiami pozwala na zbudowanie **hierarchii terminologii** dla systemu integracji wiedzy. Hierarchia terminologii ze zbioru $S_O = \{O_{x_1}, \dots, O_{x_4}, O\}$ została przedstawiona na rysunku 3.1.



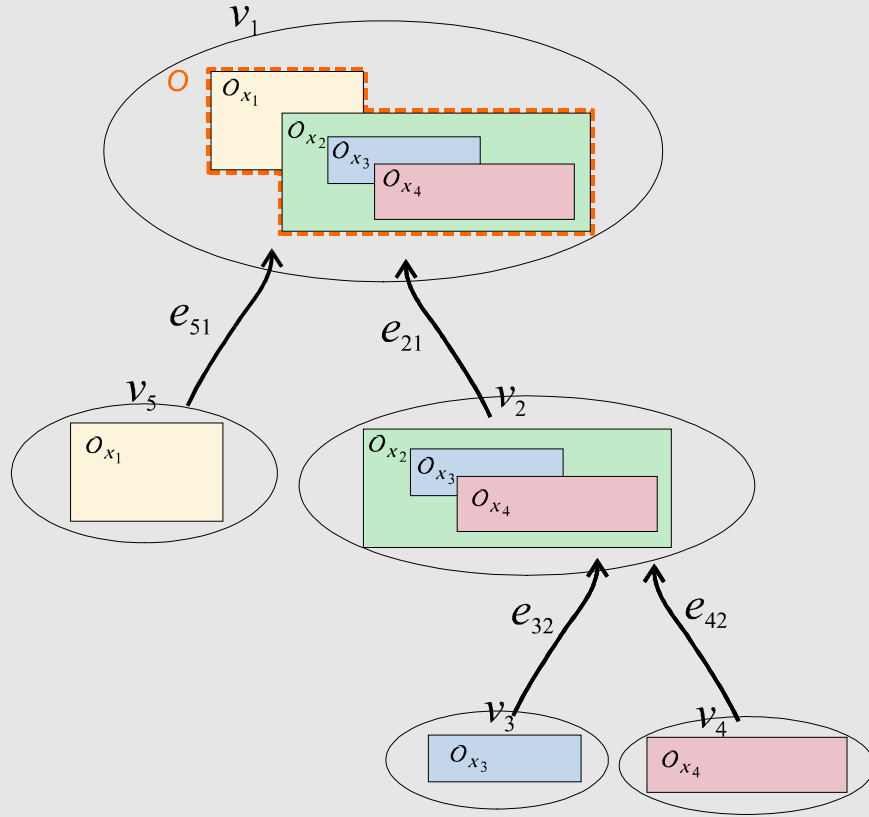
Definicja 3.6 Hierarchia terminologii $\mathcal{H}_{\mathcal{T}}$

Hierarchią terminologii $\mathcal{H}_{\mathcal{T}}$ nazywamy graf skierowany, w którym:

- 1) istnieje m węzłów, gdzie $m = |S_{\mathcal{O}}|$ jest mocą zbioru terminologii $S_{\mathcal{O}}$,
- 2) różnowartościowa **funkcja przyporządkowująca** f przyporządkowuje każdemu węzłowi v_i terminologię \mathcal{O}_{x_i} ze zbioru terminologii $S_{\mathcal{O}}$: $f(v_i) = \mathcal{O}_{x_i}$,
- 3) krawędź e_{ij} skierowana od wierzchołka v_i do v_j oznacza, że terminologia $\mathcal{O}_{x_i} = f(v_i)$ importuje terminologię $\mathcal{O}_{x_j} = f(v_j)$.

Przykład 3.2 c.d.

Przykładowa hierarchia terminologii w terminach grafowych została przedstawiona na rysunku 3.2.



Rysunek 3.2. Hierarchia terminologii w terminach grafowych

Definicja 3.7 Poprawność hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$

Hierarchia terminologii $\mathcal{H}_{\mathcal{T}}$ jest poprawna, jeśli zachodzą następujące dwa warunki:

- 1) jeśli pomiędzy dowolnymi dwoma węzłami v_i i v_j istnieje ścieżka o długości $l > 1$, to nie istnieje krawędź e_{ij} ,
- 2) dla dwóch dowolnych węzłów v_i i v_j istnieje krawędź e_{ij} lub węzeł v_k taki, że istnieją ścieżki z v_i do v_k i z v_j do v_k .

Poniżej przedstawiono algorytm budowania poprawnej hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$. Algorytm ten składa się z dwóch kroków:

1. Usunięcie cykli w hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$,
2. Usunięcie nadmiarowych krawędzi z grafu $\mathcal{H}_{\mathcal{T}}$.

Usunięcie cykli składających się z n terminologii $\{O_{x_1}, \dots, O_{x_n}\}$ w hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$ polega na przekształceniu cyklu w jedną terminologię O_x składającą się ze zbioru aksjomatów będącego sumą zbiorów aksjomatów zdefiniowanych w terminologiach O_{x_1}, \dots, O_{x_n} .

Algorytm 3.1 Algorytm usuwania cykli w hierarchii terminologii

Wejście: Hierarchia terminologii \mathcal{H}_T

Wyjście: Hierarchia terminologii \mathcal{H}_T z usuniętymi cyklami.

1. **Dopóki** \mathcal{H}_T zawiera cykl:
2. Wierzchołki cyklu v_{c_1}, \dots, v_{c_n} przepisuj na listę l_c .
3. Utwórz nowy wierzchołek v_m
4. Utwórz terminologię $O_{x_m} : Z(O_{x_m}) = Z(f(v_{c_1})) \cup \dots \cup Z(f(v_{c_n}))$
5. Przypisz terminologię O_{x_m} do wierzchołka v_m : $f(v_m) = O_{x_m}$
6. **Dla każdego** wierzchołka $v_{c_i} \in l_c$:
7. **Dla każdej** krawędzi e_{ij} z wierzchołka v_{c_i} do v_{c_j} (gdzie $v_{c_j} \notin l_c$)
8. Usuń krawędź e_{ij}
9. Dodaj krawędź e_{mj} z wierzchołka v_m do v_{c_j}
10. **Koniec** {dla każdej}
11. **Dla każdej** krawędzi e_{ji} z wierzchołka v_{c_j} do v_{c_i} (gdzie $v_{c_j} \notin l_c$)
12. Usuń krawędź e_{ji}
13. Dodaj krawędź e_{jm} z wierzchołka v_{c_j} do v_m
14. **Koniec** {dla każdej}
15. **Koniec** {dla każdego}
16. Usuń wierzchołki v_{c_1}, \dots, v_{c_n}
17. **Koniec** {dopóki}

Algorytm 3.2 Algorytm usuwania nadmiarowych krawędzi w hierarchii terminologii

Wejście: Hierarchia terminologii \mathcal{H}_T bez cykli

Wyjście: Poprawna hierarchia terminologii \mathcal{H}_T lub błąd informujący, że nie można zbudować poprawnej hierarchii terminologii \mathcal{H}_T

1. Zlicz wszystkie wierzchołki v_k , takie, że nie istnieje wierzchołek $v_l \in \mathcal{H}_T$ połączony z v_k krawędzią e_{kl}
2. **Jeżeli** liczba takich wierzchołków > 1
3. Zwróć *Błąd*: nie można zbudować poprawnej hierarchii terminologii \mathcal{H}_T
4. **Koniec** {jeżeli}
5. **Dla każdej** pary wierzchołków (v_i, v_j) (gdzie $v_i, v_j \in \mathcal{H}_T$)
6. Zlicz wszystkie ścieżki p_{ij} z v_i do v_j
7. **Jeżeli** liczba ścieżek > 1
8. **Jeżeli** istnieje krawędź e_{ij} z wierzchołka v_i do v_j
9. Usuń krawędź e_{ij}
10. **Koniec** {jeżeli}
11. **Koniec** {jeżeli}
12. **Koniec** {dla każdej}

Każdemu węzłowi grafu hierarchii terminologii jest przyporządkowany zbiór źródeł wiedzy, w których wiedza jest opisana w terminach terminologii z tego węzła w grafie. W ten sposób terminologia z danego węzła w grafie jest wzbogacona o część asercjonalną pochodzącą ze źródeł wiedzy przypisanych do tego węzła.

Dla hierarchii terminologii okreśmy relację zawierania pomiędzy dwoma terminologiami:

Definicja 3.8 Relacja zawierania dla dwóch terminologii O_{x_i} i O_{x_j} z \mathcal{H}_T

O_{x_i} zawiera się (\ll) w O_{x_j} , jeżeli istnieje ścieżka z węzła odpowiadającego terminologii O_{x_i} do węzła odpowiadającego terminologii O_{x_j} :

$$O_{x_i} \ll O_{x_j} \Leftrightarrow \exists \text{ ścieżka } p_{ij} \text{ z węzła } v_i \text{ do } v_j (f(v_i) = O_{x_i} \wedge f(v_j) = O_{x_j}).$$

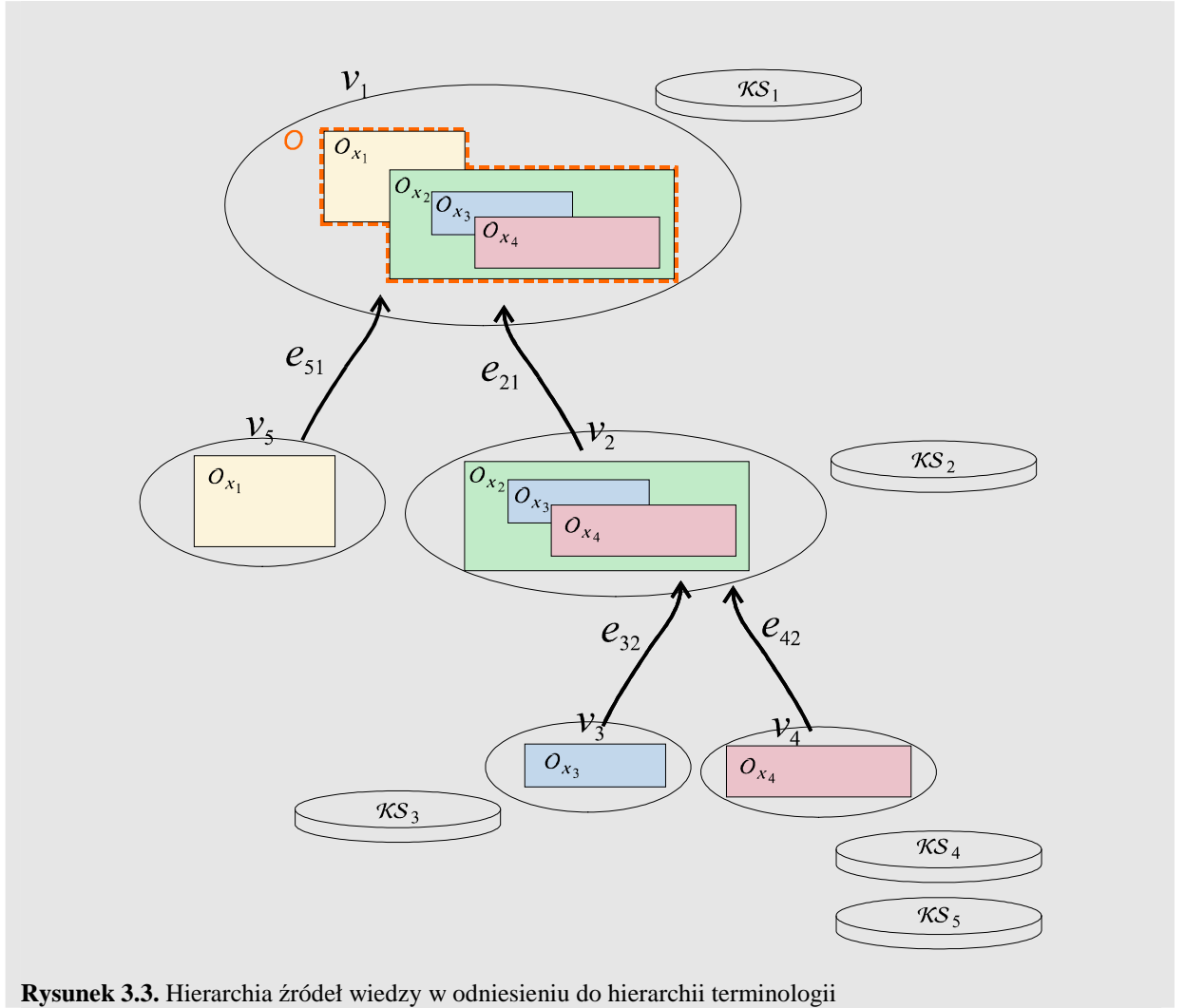
Definicja 3.9 System integracji wiedzy KISS (*Knowledge Integration Support System*)

System integracji wiedzy KISS jest parą $\langle S_O, S_{KS} \rangle$, w której $S_O = \{O_{x_1}, \dots, O_{x_m}\}$ jest zbiorem terminologii powiązanych ze sobą relacją importu, a $S_{KS} = \{KS_1, \dots, KS_n\}$ jest zbiorem źródeł wiedzy, gdzie:
 $\forall (KS = \langle \langle O_x, O_s \rangle, Q_O \rangle \in S_{KS}) (O_x \in S_O).$

Hierarchia źródeł wiedzy wynika bezpośrednio z hierarchii terminologii, w terminach których jest opisana wiedza w poszczególnych źródłach. Źródło wiedzy $KS = \langle \langle O_x, O_s \rangle, Q_O \rangle$ jest przyporządkowane do węzła v w grafie \mathcal{H}_T , dla którego:
 $f(v) = O_x \vee (Z(O_x) \in Z(f(v)) \wedge \neg \exists v_i \in \mathcal{H}_T (f(v_i) \ll f(v) \wedge Z(O_x) \in Z(f(v_i))))).$

Przykład 3.2 c.d.

Mamy pięć przykładowych źródeł wiedzy $S_{KS} = \{KS_1, KS_2, KS_3, KS_4, KS_5\}$:
 $KS_1 = \langle O, O_{s_1} \rangle$, $KS_2 = \langle O_{x_2}, O_{s_2} \rangle$, $KS_3 = \langle O_{x_3}, O_{s_3} \rangle$, $KS_4 = \langle O_{x_4}, O_{s_4} \rangle$,
 $KS_5 = \langle O_{x_4}, O_{s_5} \rangle$. Rysunek 3.3 przedstawia hierarchię źródeł wiedzy w odniesieniu do hierarchii terminologii.



Rysunek 3.3. Hierarchia źródeł wiedzy w odniesieniu do hierarchii terminologii

Niech $S_{O_s}(O_x) = \{O_s : \exists KS \in S_{KS} (KS = \langle \langle O_{x_l}, O_s \rangle, Q_O \rangle \wedge O_{x_l} \ll O_x)\}$ jest zbiorem opisów świata terminologii O_x lub terminologii zawierających się w terminologii O_x źródeł wiedzy ze zbioru S_{KS} . Zbiór $P_{O_s}(O_x) \subseteq S_{O_s}(O_x)$ jest dowolnym podzbiorem zbioru $S_{O_s}(O_x)$.

Definicja 3.10 Semantyka systemu KISS

Semantykę systemu KISS $sem(KISS)$ definiujemy jako:

$sem(KISS) = \{I(O_x, P_{O_s}(O_x)) : I \text{ jest modelem ontologii}\}$

$$O = \langle O_x, O_s \rangle, \text{ gdzie } O_s = \bigcup_{O_{s_i}} P_{O_s}(O_x) \text{ i } i=1, \dots, |P_{O_s}|.$$

Semantyka systemu KISS jest zdefiniowana jako zbiór wszystkich modeli ontologii składających się z terminologii O_x i opisu świata będącego zbiorem asercji zdefiniowanych w opisach świata pewnego podzbioru zbioru źródeł wiedzy.

Tak jak w źródłach wiedzy, tak i w systemie integracji wiedzy istnieją dwa typy zapytań: zapytania dotyczące terminologii i zapytania dotyczące opisu świata. Analogicznie jak w przypadku źródeł wiedzy, język zapytań Q_O jest sumą zbiorów zapytań dotyczących

terminologii i opisu świata. W kolejnym podrozdziale opisano język zapytań do systemu integracji wiedzy KISS.

3.2 Język zapytań do systemu integracji wiedzy

Definiując zbiór zapytań w systemie integracji wiedzy, wzorujemy się na składowej *Asks* interfejsu DIG (*Description Logic Interface*) [BECH2003]. W interfejsie DIG zostały zdefiniowane: w języku Q_{O_T} zapytania o spełnialność (ang.: *satisfiability*), o hierarchię konceptów (ang. *concept hierarchy*) i hierarchię ról (ang. *role hierarchy*), zaś w języku Q_{O_I} zapytania o osobniki.

Zapytanie $q_{O_T} \in Q_{O_T}$ jest zapytaniem dotyczącym terminologii, a proces znajdowania odpowiedzi przez system KISS jest analogiczny jak w przypadku innych systemów reprezentacji wiedzy. Zapytanie to jest zawsze zadawane do całej hierarchii terminologii \mathcal{H}_T i nie wymaga wskazania węzła odpowiedzialnego za jego wykonanie. System automatycznie wyszukuje węzeł v_i taki, że terminologia $f(v_i) = O_x$ „zna” wszystkie terminy pojawiające się w zapytaniu q_{O_T} . Wśród zapytań o osobniki Q_{O_I} w systemie KISS wyróżniamy zapytania:

- o wystąpienia konceptu (ang. *instances*),
- o typy osobnika (ang. *types*),
- o przynależność osobnika do konceptu (ang. *instance*),
- o wystąpienia ról (ang. *related individuals*),
- o dopełnienia roli (ang. *role fillers*),
- o podmioty roli (ang. *role subjects*, rozszerzenie względem interfejsu DIG) oraz
- o wartości atrybutu (ang. *told values*).

Zapytanie $q_{O_I}(O_x, P_{KS}) \in Q_{O_I}$ jest zadawane z dwoma opcjonalnymi parametrami:

O_x - terminologią, do której to zapytanie się odnosi, i zbiorem źródeł wiedzy $P_{KS} \subseteq S_{KS}$, z którego dane tworzą opis świata. Niech $P_{O_s}(P_{KS}) = \{O_s : \exists KS \in P_{KS}\}$. Odpowiedź na zapytanie $q_{O_I}(O_x, P_{KS})$ jest równa odpowiedzi na zapytanie q_{O_I} do źródła wiedzy $KS = \langle \langle O_x, O_s \rangle, Q_O \rangle$, gdzie $O_s = \bigcup_{O_{s_i}} P_{O_{s_i}}(P_{KS})$ i $i=1, \dots, |P_{O_s}(P_{KS})|$.

Definicja 3.11 Semantyka zapytania $q_{O_I}(O_x, P_{KS}) \in Q_{O_I}$ w systemie KISS

Odpowiedź $r_{O_I}(O_x, P_{KS})^{KISS}$ na zapytanie $q_{O_I}(O_x, P_{KS})$ do KISS jest zbiorem następujących krotek:

$$r_{O_I}(O_x, P_{KS})^{KISS} = \{r : \forall I(O_x, P_{KS}), I \in sem(KISS) (r \in r_{O_I}(O_x, P_{KS})^I)\}$$

Krotki r stanowiące odpowiedź na zapytanie są zależne od typu zapytania. Przykładowo, w zapytaniu o wystąpienia roli będą to krotki dwuelementowe składające się z nazw osobników, zaś w zapytaniu o przynależność osobnika do konceptu będzie to zawsze jedna krotka o wartości *true* (osobnik przynależy do konceptu), *false* (osobnik nie przynależy do konceptu) lub *maybe* (nie można jednoznacznie stwierdzić, czy osobnik przynależy do konceptu czy też nie).

Definicja 3.12

Poprawność (ang. *soundness*) odpowiedzi $r_{O_I}(O_x, P_{KS})^{KISS}$ na zapytanie $q_{O_I}(O_x, P_{KS})$

Odpowiedź $r_{O_I}(O_x, P_{KS})^{KISS}$ na zapytanie $q_{O_I}(O_x, P_{KS})$ do KISS jest poprawna,

jeżeli:

$$\forall r(r \in r_{O_I}(O_x, P_{KS})^{KISS}) \Rightarrow \forall I(O_x, P_{KS}), I \in sem(KISS) (r \in r_{O_I}(O_x, P_{KS})^I)$$

Definicja 3.13 Kompletność (ang. *completeness*) odpowiedzi $r_{O_I}(O_x, P_{KS})^{KISS}$ na zapytanie $q_{O_I}(O_x, P_{KS})$

Odpowiedź $r_{O_I}(O_x, P_{KS})^{KISS}$ na zapytanie $q_{O_I}(O_x, P_{KS})$ do KISS jest kompletna, jeżeli:

$$\neg \exists r, \forall I(O_x, P_{KS}), I \in sem(KISS) (r \in r_{O_I}(O_x, P_{KS})^I \wedge r \notin r_{O_I}(O_x, P_{KS})^{KISS})$$

Parametry w zapytaniu $q_{O_I}(O_x, P_{KS})$ są opcjonalne. Brak tylko parametru pierwszego oznacza, że terminologia O_x jest terminologią spełniającą poniższe warunki:

1. O_x zna wszystkie terminy zdefiniowane w zapytaniu $q_{O_I}(O_x, P_{KS})$,
2. $\forall O_{x_l} \in \{O_{x_l} : \exists KS = \langle \langle O_{x_l}, O_s \rangle, Q_O \rangle \in P_{KS}\} (O_{x_l} \ll O_x)$,
3. $\neg \exists O_{x_l} \in S_O$ spełniająca warunki 1. i 2. ($O_{x_l} \ll O_x$).

Brak tylko parametru drugiego oznacza, że

$$P_{KS} = \{KS : KS = \langle \langle O_{x_l}, O_s \rangle, Q_O \rangle \wedge O_{x_l} \ll O_x\}.$$

Brak obu parametrów oznacza, że:

1. $O_x : \neg \exists O_{x_l} (O_x \ll O_{x_l})$,
2. $P_{KS} = S_{KS}$.

Przykład 3.2 c.d.

Przyjmijmy, że budujemy system KISS dla naszej przykładowej hierarchii terminologii, w której:

O_{x4}

$Pracownik \sqsubseteq \exists jestNaStanowisku.Stanowisko$

$\exists jestNaStanowisku.\top \sqsubseteq Pracownik$

$\exists jestNaStanowisku.\neg Stanowisko \sqsubseteq \perp$

$Stanowisko \sqcap Pracownik \equiv \perp$

O_{x3}

$Kobieta \sqcup Mężczyzna \equiv Człowiek$

$Kobieta \sqcap Mężczyzna \equiv \perp$

O_{x2}

$Pracownik \sqsubseteq Człowiek$

$Człowiek \sqsubseteq Ssak$

$Ssak \sqcap Stanowisko \equiv \perp$

$Kobieta \sqcup Mężczyzna \equiv Człowiek$ (z O_{x3})

$Kobieta \sqcap Mężczyzna \equiv \perp$ (z O_{x3})

$Pracownik \sqsubseteq \exists \text{jestNaStanowisku}.Stanowisko$ (z O_{x4})

$\exists \text{jestNaStanowisku}.\top \sqsubseteq Pracownik$ (z O_{x4})

$\exists \text{jestNaStanowisku}.\neg Stanowisko \sqsubseteq \perp$ (z O_{x4})

O_{x1}

$ZwierzętaRoślinożerne \sqsubseteq \exists \text{zjada}.Rośliny \wedge \exists \text{zjada}.Rośliny$

O

$ZwierzętaRoślinożerne \sqsubseteq \exists \text{zjada}.Rośliny \wedge \exists \text{zjada}.Rośliny$ (z O_{x1})

$Pracownik \sqsubseteq Człowiek$ (z O_{x2})

$Człowiek \sqsubseteq Ssak$ (z O_{x2})

$Kobieta \sqcup Mężczyzna \equiv Człowiek$ (z O_{x3})

$Kobieta \sqcap Mężczyzna \equiv \perp$ (z O_{x3})

$Pracownik \sqsubseteq \exists \text{jestNaStanowisku}.Stanowisko$ (z O_{x4})

$\exists \text{jestNaStanowisku}.\top \sqsubseteq Pracownik$ (z O_{x4})

$\exists \text{jestNaStanowisku}.\neg Stanowisko \sqsubseteq \perp$ (z O_{x4})

Źródła wiedzy zawierają kolejno następujące asercje:

\mathcal{KS}_1

$Roślinożerne(\text{ChipWiewiórka})$

$Roślinożerne(\text{DaleWiewiórka})$

\mathcal{KS}_2

$Ssak(\text{ChipWiewiórka})$

$Ssak(\text{DaleWiewiórka})$

$Pracownik(\text{Linguini})$

\mathcal{KS}_3

$Mężczyzna(\text{Linguini})$

\mathcal{KS}_4

$\text{jestNaStanowisku}(\text{Linguini}, \text{Kucharz})$

\mathcal{KS}_5

$\text{jestNaStanowisku}(\text{Kopciuszek}, \text{Sprzątaczką})$

Przeanalizujemy kilka zapytań do przykładowego systemu integracji wiedzy:

- 1) $instances(ZwierzętaRoślinożerne)(O)$
- 2) $instances(ZwierzętaRoślinożerne)(O, \{KS_5\})$
- 3) $instance(Kopciuszek, Ssak)(\{KS_5\})$

Odpowiedź na zapytanie (1) to: ChipWiewiórka, DaleWiewiórka. Dane są pobierane ze wszystkich źródeł, ponieważ terminologia O importuje wszystkie pozostałe terminologie ze zbioru S_O . W zapytaniu (2) zbiór źródeł wiedzy, do których zapytanie ma być zadane, jest ściśle określony. W źródle KS_5 nie ma żadnego osobnika będącego zwierzęciem roślinożernym. Ostatnie zapytanie to zapytanie o to, czy Kopciuszek jest *Ssakiem*. Ze względu na to, że terminologia O_{x_2} jest najniżej w hierarchii terminologii z terminologii znających termin *Ssak* i ponieważ źródło KS_5 jest opisane terminologią $O_{x_4} \ll O_{x_2}$, to zapytanie $instance(Kopciuszek, Ssak)(\{KS_5\})$ możemy sformułować jako $instance(Kopciuszek, Ssak)(O_{x_2}, \{KS_5\})$. Odpowiedź na to zapytanie jest *true*. Kopciuszek jest na stanowisku, więc jest pracownikiem. Każdy pracownik jest człowiekiem, a każdy człowiek ssakiem.

3.3 Podsumowanie

W tym rozdziale przedstawiono model systemu integracji wiedzy oraz zdefiniowano język zapytań do systemu integracji wiedzy. Tym samym cel pierwszy rozprawy: przedstawienie modelu integracji wiedzy, został osiągnięty. W tym rozdziale przedstawiono model formalny systemu. W rozdziale 6 przedstawiono konkretne metody opisywania zależności między terminologiami i algorytmy przetwarzania zapytań w systemie integracji wiedzy.

Rozdział 4. Kartograficzna reprezentacja wiedzy podstawą algorytmów systemu integrującego wiedzę

4.1 Kartograficzna metoda reprezentacji wiedzy

Główne zadania systemu zarządzania wiedzą obejmują:

1. Przechowywanie ontologii w postaci łatwo dostępnej bazy wiedzy.
2. Udostępnianie mechanizmu aktualizacji przechowywanej informacji poprzez dodawanie nowych faktów w postaci asercji.
3. Udzielanie odpowiedzi na zapytania dotyczące przechowywanej wiedzy i przeprowadzanie procesu wnioskowania, w celu uwzględnienia w odpowiedziach na zapytania wiedzy nie wyrażonej jawnie.

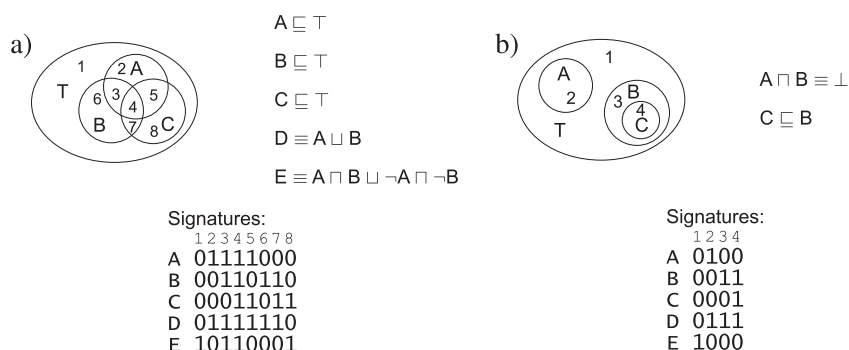
Podejście kartograficzne zostało opracowane w celu usprawnienia wykonywania powyższych zadań. Podejście to opiera się na trzech głównych spostrzeżeniach, słusznych dla stosunkowo szerokiego zakresu zastosowań praktycznych:

1. Terminologiczna część bazy wiedzy jest aktualizowana stosunkowo rzadko, może więc być uznana za stałą.
2. Do bazy wiedzy kierowane są znacznie częściej zapytania niż nowe asercje. Zatem czas odpowiadania na zapytania jest kluczowy, podczas gdy czas aktualizacji jest mniej ważny.

Kartograficzna metoda reprezentacji wiedzy została zaproponowana po raz pierwszy przez Wojciecha Waloszka i jej opis formalny oraz aktualny etap rozwoju został opisany między innymi w [WALO2008], [GOCZ2007c], [GOCZ2006d].

W reprezentacji tej każdy koncept ma przypisaną *sygnaturę*. Sygnatura jest tablicą bitów reprezentujących obszary atomowe na mapie konceptów. W rzeczywistości mapa konceptów pokazuje zależności pomiędzy konceptami w terminologii i może być reprezentowana graficznie w formie podobnej do diagramów Venna. Przykładowa mapa konceptów dla pewnej ontologii została przedstawiona na rysunku 4.1. Każdy obszar atomowy (nazywany dalej regionem), tj. obszar nie zawierający żadnego innego obszaru, reprezentuje unikalne, poprawne przecięcie konceptów atomowych. Poprzez poprawne przecięcie rozumiemy takie przecięcie, które jest spełnialne w odniesieniu do zadanej terminologii. Przecięcia konceptów, które nie są dozwolone przez aksjomaty terminologiczne, są usuwane z mapy. Na mapie znajduje się n regionów, a do każdego z nich jest przypisana kolejna liczba z zakresu $\langle 1, n \rangle$. Ponieważ każdy obszar na mapie składa się z pewnej liczby regionów, może być reprezentowany jako ciąg bitów o długości n , przy czym bit na i -tej pozycji jest ustawiony wtedy i tylko wtedy, gdy i -ty region jest zawarty w danym obszarze. W przeciwnym razie bit ten jest wyzerowany. Każdemu konceptowi jest przypisany pewien obszar, co jest równoważne przypisaniu mu odpowiedniego ciągu bitów, zwanego właśnie sygnaturą.

W terminach sygnatur możemy opisać dowolną kombinację dopełnienia, sumy i przecięcia konceptów przez odwzorowanie tych operacji w działania algebry Boole'a.



Rysunek 4.1. Przykładowa mapa konceptów (a), po wprowadzeniu nowych aksjomatów (b)

Analogicznie, w procesie wczytywania asercji o osobnikach, każdemu z nich jest nadawana **sygnatura osobnicza**. Różnica między sygnaturami osobniczymi a sygnaturami konceptów polega na tym, że bit ustawiony w sygnaturze konceptu oznacza, że odpowiedni region należy do obszaru przypisanego danemu konceptowi, natomiast w przypadku sygnatur osobniczych bit ustawiony oznacza jedynie, że osobnik może przynależeć do danego regionu. W rzeczywistości każdy osobnik należy do jednego i tylko jednego regionu. Jeśli więc sygnatura osobnicza zawiera więcej niż jeden ustawiony bit, oznacza to, że nie można jednoznacznie określić, do którego z tych regionów dany osobnik należy.

Do systemu wnioskującego wczytywane są również wystąpienia ról. Podstawą procesu wnioskowania w systemach opartych na kartograficznej reprezentacji wiedzy jest porównywanie sygnatur. Przykładowo problem określenia zbioru wystąpień konceptu C jest ograniczony do znalezienia wszystkich osobników, których sygnatury są zawarte w sygnaturze konceptu C (sygnatura s_1 jest zawarta w sygnaturze s_2 , jeśli każdy bit sygnatury s_1 jest nie większy niż odpowiadający mu bit sygnatury s_2).

W punkcie 4.2 zostanie przedstawiony formalizm metody kartograficznej, w zakresie podstawowym wykorzystanym w kolejnych rozdziałach tej rozprawy oraz ogólna koncepcja algorytmów wnioskujących z wiedzy reprezentowanej kartograficznie, zaś w punkcie 4.3 opis systemu KaSeA jako źródła wiedzy w systemie integracji wiedzy. Formalizm kartografii wiedzy oraz opis algorytmów wnioskujących z wiedzy reprezentowanej kartograficznie został zaczerpnięty z [WALO2008].

4.2 Podstawowy formalizm kartograficznej reprezentacji wiedzy

Podstawowym terminem w kartografii wiedzy jest pojęcie sygnatury.

Definicja 4.1 Sygnatura długości n

Sygnaturą o długości n , $n \in \mathbb{N}$, nazywamy element zbioru $\mathbb{B}^n = \{0,1\}^n$. Sygnatura s o długości n jest ciągiem n -elementowym.

W poniższych definicjach przyjęto oznaczenie s_i na i -ty bit sygnatury s .

Definicja 4.2 Działania na sygnaturach

Sumą sygnatur s i t długości n nazywamy działanie binarne $\vee: \mathbb{B}^n \times \mathbb{B}^n \rightarrow \mathbb{B}^n$ zdefiniowane następująco:

$[\vee(s, t)]_i = s_i \vee t_i, \quad i \in [1, n]$, gdzie \vee jest operacją logicznej sumy na pojedynczych bitach.

Przecięciem sygnatur s i t o długości n nazywamy działanie binarne $\wedge: \mathbb{B}^n \times \mathbb{B}^n \rightarrow \mathbb{B}^n$ zdefiniowane następująco: $[\wedge(s, t)]_i = s_i \wedge t_i, \quad i \in [1, n]$, gdzie \wedge jest operacją logicznego iloczynu na pojedynczych bitach.

Negacją sygnatury s o długości n nazywamy działanie unarne $\sim: \mathbb{B}^n \rightarrow \mathbb{B}^n$ zdefiniowane następująco: $[\sim(s)]_i = \neg s_i, \quad i \in [1, n]$, gdzie \neg jest operacją negacji pojedynczego bitu.

Definicja 4.3 Porządek w zbiorze sygnatur

Relacją podrzędności sygnatur długości n nazywamy relację binarną $\leq \subseteq \mathbb{B}^n \times \mathbb{B}^n$ zdefiniowaną następująco: $s \leq t \Leftrightarrow \forall i \in [1, n]: s_i \leq t_i$, gdzie \leq jest standardowo zdefiniowaną relacją mniejszości-równości. Relacją ścisłej podrzędności nazywamy relację binarną $< \subseteq \mathbb{B}^n \times \mathbb{B}^n$ zdefiniowaną następująco: $s < t \Leftrightarrow s \leq t \wedge s \neq t$.

Kartograficzna reprezentacja wiedzy może być aktualnie stosowana do dialektu kartograficznego \mathcal{CLC} .

Definicja 4.4 Dialekt kartograficzny \mathcal{CLC}

Dialektem kartograficznym \mathcal{CLC} dla terminologii T , oznaczany $\mathcal{CLC}(T)$, nazywamy dialekt logiki opisowej, w którym możliwe do sformułowania są następujące koncepty:

- 1) koncept prosty A zawarty w terminologii T ,
 - 2) kwantyfikacja egzystencjalna $\exists R.C$ zdefiniowana w terminologii T ,
 - 3) kwantyfikacja ogólna $\forall R.C$,
 - 4) koncept uniwersalny,
 - 5) koncept pusty,
 - 6) przecięcie konceptów,
 - 7) suma konceptów,
 - 8) dopełnienie konceptu.
-

Definicja 4.5 Mapa konceptów dla dialektu $\mathcal{CLC}(T)$

Mapą konceptów dla dialektu $\mathcal{CLC}(T)$ nazywamy funkcję $s: C(\mathcal{CLC}(T)) \rightarrow \mathbb{B}^n$ przypisującą każdemu konceptowi wyrażonemu w $\mathcal{CLC}(T)$ sygnaturę o długości n i mającą następującą właściwość ($C, D \in C(\mathcal{CLC}(T))$):

$$s(C) \leq s(D) \Leftrightarrow T \models C \sqsubseteq D$$

Liczbę n ($n \in \mathbb{N}$) nazywamy rozmiarem mapy konceptów.

Definicja 4.6 Rozmieszczenie osobników bazy wiedzy K

Dla danej bazy wiedzy $K = (T, A)$ wyrażonej w dialekcie $C\mathcal{L}C(T)$ rozmieszczeniem osobników na mapie konceptów nazywamy funkcję $s: A_I \rightarrow \mathbb{B}^n$ posiadającą następującą właściwość ($a \in A_I, C \in C(C\mathcal{L}C(T))$):
 $s(a) \leq s(C) \Leftrightarrow K \models C(a)$
 gdzie A_I jest zbiorem nazw osobników, a n jest rozmiarem mapy konceptów s .

Algorytmy wnioskujące wykorzystują operacje na sygnaturach określone w definicji 4.3. Poniżej przedstawiono wykorzystanie metody kartograficznej dla rozwiązania problemów wnioskowania z terminologii i o osobnikach wypunktowanych w rozdziale 2 niniejszej rozprawy:

- 1) **problem zawierania** (ang. *subsumption problem*) – czy zbiór wystąpień konceptu C jest zawsze podzbiorem wystąpień konceptu D ? (Czy $C \sqsubseteq D$?):
 $C \sqsubseteq D \Leftrightarrow s(C) \leq s(D)$;
- 2) **problem spełnialności** (ang. *satisfiability problem*) – czy koncept C może mieć wystąpienia (Czy $C \equiv \perp$?):
 $C \equiv \perp \Leftrightarrow s(C) = s(\perp)$, gdzie $s(\perp)$ jest sygnaturą, której każdy bit jest ustawiony na wartość 0;
- 3) **problem równoważności** (ang. *equivalence problem*) – czy zbiory wystąpień konceptów C i D są zawsze równe? (Czy $C \equiv D$?):
 $C \equiv D \Leftrightarrow s(C) = s(D)$;
- 4) **problem rozłączności** (ang. *disjointness problem*) – czy zbiory konceptów C i D są zawsze rozłączne? (Czy $C \sqcap D \equiv \perp$?):
 $C \sqcap D \equiv \perp \Leftrightarrow s(C) \wedge s(D) = s(\perp)$;
- 5) **problem określenia zbioru wystąpień konceptu** (ang. *retrieval problem*) – jakie osobniki należą do konceptu C ?
 $i^I \in C^I \Leftrightarrow s(i) \leq s(C)$;
- 6) **problem sprawdzenia przynależności** (ang. *instance check problem*) – czy dany osobnik należy do konceptu C ?
 $i^I \in C^I \Leftrightarrow s(i) \leq s(C)$;
- 7) **problem sprawdzenia spójności** (ang. *consistency check*) – czy istnieje niepusty model bazy wiedzy?
 baza wiedzy jest spójna $\Leftrightarrow (\forall (i^I \in C^I) (s(i) \neq s(\perp))) \wedge (\forall C (s(C) \neq s(\perp)))$

4.3 Źródło wiedzy *Knowledge Signature Analyser* – KaSeA

System *Knowledge Signature Analyser* – KaSeA został zaimplementowany jako baza wiedzy dla systemu PIPS (ang. *Personailised Information Platform for Life and Heath Services*) [PIPS2007]. System KaSeA dostarcza mechanizmów wnioskowania za pomocą interfejsu *Description Logic Interface by Gdańsk University of Technology* – DIGUT [DIGUT] opartego na interfejsie DIG 1.1. [BECH2003]. KaSeA dostarcza mechanizmów

wnioskowania dla dialektu ($CLC(T)$), które można zastosować do rozwiązania problemów wnioskowania wymienionych powyżej. Problem sprawdzenia spójności bazy wiedzy nie jest zaimplementowany wprost i wymaga sprawdzenia dla każdego konceptu, czy $C \sqsubseteq \perp$ i dla każdego osobnika, czy $i^I \in \perp^I$).

Zgodnie z definicją 3.3 źródłem wiedzy $\mathcal{KS} = \langle O, Q_O \rangle$ nazywamy system opisujący wiedzę w terminach ontologii O i dostarczający metod wnioskowania umożliwiających poprawne i kompletne odpowiadanie na zapytania sformułowane w języku zapytań Q_O . Model systemu integracji wiedzy zakłada, że źródło wiedzy umożliwia poprawne i kompletne odpowiadanie na zapytania sformułowane w języku zapytań Q_O . Po przeanalizowaniu dostępnych systemów wnioskujących z wiedzy ontologicznej wybrano na źródło wiedzy system KaSeA. System KaSeA nie gwarantuje kompletności odpowiedzi na zapytania asercjonalne², jakkolwiek:

- wydajność algorytmów wnioskujących w systemie KaSeA jest lepsza aniżeli w systemach konkurencyjnych.
- przypadki braku kompletności odpowiedzi są nieliczne i nie eliminują możliwości praktycznego zastosowania kartograficznej reprezentacji wiedzy i systemu KaSeA,

4.4 Podsumowanie

W pracy doktorskiej [WALO2008] zostało dowiedzione, że przez zastosowanie zaproponowanych w rozprawie metod strukturalnej analizy ontologii możliwe jest polepszenie parametrów jakościowych i ilościowych systemów zarządzania wiedzą dla szerokiej klasy ontologii spotykanych w praktyce. Wskazuje na to również zastosowanie systemu KaSeA w systemie PIPS [PIPS2008] zarządzającym wiedzą medyczną. Dlatego w niniejszej rozprawie przyjęto, że również wiedza w systemie integracji wiedzy będzie reprezentowana kartograficznie. Potencjał metody wskazuje, że jej ograniczenia prawdopodobnie zostaną usunięte, a ukierunkowanie metody na szybkość odpowiadania na zapytania wskazuje, że wydajność metody umożliwia jej szerokie praktyczne wykorzystanie.

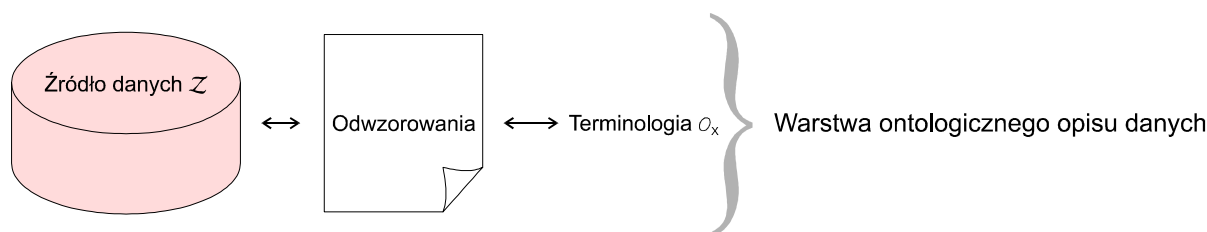
² Systemy wnioskujące oparte na kartograficznej reprezentacji wiedzy nie gwarantują kompletności odpowiedzi na zapytania asercjonalne. Jest to spowodowane niedostatkami tej reprezentacji na aktualnym poziomie jej rozwoju. W ramach grupy KMG [KMG] cały czas trwają prace mające na celu rozszerzeniem kartograficznej reprezentacji wiedzy o nowe typy sygnatur umożliwiających dostarczenie mechanizmów wnioskowania gwarantujących kompletność odpowiedzi na zapytania asercjonalne.

Rozdział 5. Metoda tworzenia źródeł wiedzy przez semantyczne wzbogacanie źródeł danych

5.1 Ontologiczny opis danych

Metoda tworzenia źródeł wiedzy poprzez semantyczne wzbogacanie źródeł danych wymaga zbudowania warstwy wiedzy opisującej źródło danych. Przez **źródło danych** rozumiemy miejsce składowania danych, w którym są one przechowywane i mogą być z niego pobierane. Definicja 3.1 wprowadzająca pojęcie źródła wiedzy wymaga opisanie wiedzy w terminach ontologii O . Metoda SED (*Semantic Enrichment of Data*) umożliwia wzbogacenie źródła danych \mathcal{Z} o dodatkowe informacje pozwalające na traktowanie danych jako pewnego zbioru asercji. Niniejszy rozdział opisuje metodę SED, przeplatając jej opis formalny przykładem wiodącym.

Na rysunku 5.1 pokazano, o jakie elementy metoda SED wzbogaca dane składowane w źródle danych, aby możliwe było traktowanie tych danych jako zbioru asercji.



Rysunek 5.1. Dane wzbogacone o informacje umożliwiające traktowanie ich jako asercji

Terminologia O_x definiuje zbiór terminów, które są używane do opisanie semantyki źródła danych \mathcal{Z} . **Odwzorowania** opisują dane z zewnętrznego źródła danych w terminach określonych w terminologii O_x .

Przykład 5.1

Przyjmijmy, że mamy pewne relacyjne źródło danych \mathcal{Z} . Schemat relacyjny tego źródła został opisany poniżej w postaci skryptu SQL:

```
CREATE TABLE STANOWISKO (
  IDSTANOWISKA NUMBER(10),
  NAZWA VARCHAR2(256),
  WYNAGRODZENIE NUMBER(10, 2),
  PRIMARY KEY(IDSTANOWISKA)
);
```

```
CREATE TABLE PRACOWNIK (
  NIP VARCHAR2(11),
  IMIE VARCHAR2(256),
  NAZWISKO VARCHAR(1024),
  IDSTANOWISKA NUMBER(10),
  PLEC CHAR(1),
  PRIMARY KEY(NIP),
  FOREIGN KEY (IDSTANOWISKA) REFERENCES STANOWISKO (IDSTANOWISKA)
);
```

W obu tabelach znajduje się kilka krotek. W tabeli STANOWISKO:

IDSTANOWISKA	NAZWA	WYNAGRODZENIE
1	Informatyk	4000
2	Prezes	7500
3	Administracja	1500

a w tabeli PRACOWNIK:

NIP	IMIE	NAZWISKO	IDSTANOWISKA	PLEC
51223490121	Jan	Kowalski	1	M
53490212634	Anna	Pietrzak	1	K
59912340978	Tomasz	Głabecki	1	M
56767674510	Witold	Jaworski	2	M
57601293847	Marta	Grabowska	3	K
54345546900	Ilona	Dylewska	3	K

Mamy również określoną pewną terminologię O_x , w terminach której chcemy opisać semantykę danych zawartych w źródle \mathcal{Z} . Niech terminologią tą będzie terminologia O_x ontologii pracowników zdefiniowana w przykładzie 3.1.

Odwzorowania są zapisane w postaci zapytań SQL do serwera relacyjnych baz danych:

Kobieta = select NIP from PRACOWNIK where PLEC = 'K'

Mężczyzna = select NIP from PRACOWNIK where PLEC = 'M'

Pracownik = select NIP from PRACOWNIK

Stanowisko = select NAZWA || '_' || IDSTANOWISKA from STANOWISKO

jestNaStanowisku = select NIP, NAZWA || '_' || IDSTANOWISKA from PRACOWNIK,
STANOWISKO where PRACOWNIK.IDSTANOWISKA =
STANOWISKO.IDSTANOWISKA

dostajeWynagrodzenie = select NIP, WYNAGRODZENIE from PRACOWNIK, STANOWISKO
where PRACOWNIK.IDSTANOWISKA = STANOWISKO.IDSTANOWISKA

Korzystając z terminologii O_x , zdefiniujemy ontologię $O = \langle O_x, O_s \rangle$ dla źródła danych \mathcal{Z} . Terminologia O_x jest wyrażona w terminach alfabetu $\mathcal{A}_{O_T} = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A}$. Asercje dotyczące osobników O_s są wyrażone w terminach alfabetu $\mathcal{A}_O = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A} \cup \mathcal{A}_{O_I}$, w którym alfabet \mathcal{A}_{O_C} , \mathcal{A}_{O_R} i \mathcal{A}_{O_A} zostały już określone w \mathcal{A}_{O_T} . Zanim zostanie określony alfabet \mathcal{A}_{O_I} , wprowadzimy definicję odwzorowania. Istnieją trzy typy odwzorowań: *odwzorowanie konceptu*, *odwzorowanie roli* i *odwzorowanie atrybutu*.

Definicja 5.1 Odwzorowanie konceptu

Odwzorowaniem konceptu $\mathcal{M}_C = \langle C, q_C \rangle$ nazywamy parę składającą się z konceptu C , atomowego lub złożonego, wyrażonego w logice opisowej \mathcal{L}_O na alfabecie $\mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A}$, oraz zapytania q_C do źródła danych \mathcal{Z} , dla którego odpowiedzią $r_C = \{i: i \text{ jest nazwą osobnika}\}$ jest zbiór nazw osobników.

Przykład 5.1 c.d.

W naszym przykładzie mamy cztery odwzorowania konceptów:

$\mathcal{M}_{Kobieta} = \langle Kobieta, q_{Kobieta} \rangle$,

gdzie $q_{Kobieta} = \text{select NIP from PRACOWNIK where PLEC} = 'K'$

$\mathcal{M}_{Mężczyzna} = \langle Mężczyzna, q_{Mężczyzna} \rangle$,

gdzie $q_{Mężczyzna} = \text{select NIP from PRACOWNIK where PLEC} = 'M'$

$\mathcal{M}_{Pracownik} = \langle Pracownik, q_{Pracownik} \rangle$,

gdzie $q_{Pracownik} = \text{select NIP from PRACOWNIK}$

$\mathcal{M}_{Stanowisko} = \langle Stanowisko, q_{Stanowisko} \rangle$,

gdzie $q_{Stanowisko} = \text{select NAZWA} || '_' || \text{IDSTANOWISKA from STANOWISKO}$

Odpowiedzi dla poszczególnych zapytań są następującymi zbiorami:

$r_{Kobieta} = \{53490212634, 57601293847, 54345546900\}$.

$r_{Mężczyzna} = \{51223490121, 59912340978, 56767674510\}$,

$r_{Pracownik} = \{51223490121, 53490212634, 59912340978, 56767674510, 57601293847, 54345546900\}$,

$r_{Stanowisko} = \{\text{Informatyk}_1, \text{Prezes}_2, \text{Administracja}_3\}$.

Definicja 5.2 Odwzorowanie roli

Odwzorowaniem roli $\mathcal{M}_R = \langle R, q_R \rangle$ nazywamy parę składającą się z roli R , atomowej lub złożonej, wyrażonej w logice opisowej \mathcal{L}_O na alfabecie \mathcal{A}_{O_R} oraz zapytania q_R do źródła danych \mathcal{Z} , dla którego odpowiedzią $r_R = \{ \langle i, j \rangle : i, j \text{ są nazwami osobników} \}$ jest zbiór par nazw osobników.

Przykład 5.1 c.d.

W naszym przykładzie mamy jedno odwzorowanie roli:

$\mathcal{M}_{jestNaStanowisku} = \langle jestNaStanowisku, q_{jestNaStanowisku} \rangle$,

gdzie $q_{jestNaStanowisku} = \text{select NIP, NAZWA} || '_' || \text{IDSTANOWISKA from PRACOWNIK, STANOWISKO where PRACOWNIK.IDSTANOWISKA} = \text{STANOWISKO.IDSTANOWISKA}$

Odpowiedź dla zapytania $q_{jestNaStanowisku}$ została określona poniżej:

$r_{jestNaStanowisku} = \{ \langle 51223490121, \text{Informatyk}_1 \rangle, \langle 53490212634, \text{Informatyk}_1 \rangle, \langle 59912340978, \text{Informatyk}_1 \rangle, \langle 56767674510, \text{Prezes}_2 \rangle, \langle 57601293847, \text{Administracja}_3 \rangle, \langle 54345546900, \text{Administracja}_3 \rangle \}$.

Definicja 5.3 Odwzorowanie atrybutu

Odwzorowaniem atrybutu $\mathcal{M}_A = \langle A, q_A \rangle$ nazywamy parę składającą się z atrybutu A wyrażonego w logice opisowej \mathcal{L}_O na alfabecie \mathcal{A}_{O_A} oraz zapytania q_A do źródła danych \mathcal{Z} , dla którego odpowiedzią $r_A = \{ \langle i, v \rangle : i \text{ jest nazwą osobnika, a } v \text{ pewną wartością} \}$ jest zbiór par: nazwa osobnika, wartość.

Przykład 5.2 Odwzorowania atrybutów

W naszym przykładzie mamy jedno odwzorowanie atrybutu:

$\mathcal{M}_{\text{dostajeWynagrodzenie}} = \langle \text{dostajeWynagrodzenie}, q_{\text{dostajeWynagrodzenie}} \rangle$,

gdzie $q_{\text{dostajeWynagrodzenie}} = \text{select NIP, WYNAGRODZENIE from PRACOWNIK, STANOWISKO where PRACOWNIK.IDSTANOWISKA = STANOWISKO.IDSTANOWISKA}$

Odpowiedź dla zapytania $q_{\text{dostajeWynagrodzenie}}$ została określona poniżej:

$r_{\text{dostajeWynagrodzenie}} = \{ \langle 51223490121, 4000 \rangle, \langle 53490212634, 4000 \rangle, \langle 59912340978, 4000 \rangle, \langle 56767674510, 7500 \rangle, \langle 57601293847, 1500 \rangle, \langle 54345546900, 1500 \rangle \}$.

Źródło danych jest opisane pewnym **zbiorem odwzorowań**.

Definicja 5.4 Zbiór odwzorowań \mathcal{S}

Zbiór odwzorowań $\mathcal{S} = \mathcal{S}_{\mathcal{M}_C} \cup \mathcal{S}_{\mathcal{M}_R} \cup \mathcal{S}_{\mathcal{M}_A}$ definiujemy jako sumę zbiorów: odwzorowań konceptów $\mathcal{S}_{\mathcal{M}_C}$, odwzorowań ról $\mathcal{S}_{\mathcal{M}_R}$ i odwzorowań atrybutów $\mathcal{S}_{\mathcal{M}_A}$.

Przykład 5.1 c.d. Zbiór odwzorowań

W naszym przykładzie zbiór odwzorowań $\mathcal{S} = \mathcal{S}_{\mathcal{M}_C} \cup \mathcal{S}_{\mathcal{M}_R} \cup \mathcal{S}_{\mathcal{M}_A}$, gdzie:

$\mathcal{S}_{\mathcal{M}_C} = \{ \mathcal{M}_{\text{Kobieta}}, \mathcal{M}_{\text{Mężczyzna}}, \mathcal{M}_{\text{Pracownik}}, \mathcal{M}_{\text{Stanowisko}} \}$,

$\mathcal{S}_{\mathcal{M}_R} = \{ \mathcal{M}_{\text{jestNaStanowisku}} \}$,

$\mathcal{S}_{\mathcal{M}_A} = \{ \mathcal{M}_{\text{dostajeWynagrodzenie}} \}$.

Po zdefiniowaniu odwzorowań możliwe jest określenie alfabetu \mathcal{A}_{O_I} będącego zbiorem nazw osobników występujących w odpowiedziach na zapytania q_C , q_R , q_A zdefiniowane w zbiorze odwzorowań.

$\mathcal{A}_{O_I} = \{ i : ((\exists \mathcal{M}_C \in \mathcal{S}_{\mathcal{M}_C} (i \in r_C)) \vee (\exists k \exists \mathcal{M}_R \in \mathcal{S}_{\mathcal{M}_R} (\langle i, k \rangle \in r_R \vee \langle k, i \rangle \in r_R)) \vee (\exists \mathcal{M}_A \in \mathcal{S}_{\mathcal{M}_A} (\langle i, v \rangle \in r_A))) \}$.

Przykład 5.1 cd. Alfabet \mathcal{A}_{O_I} dla ontologii \mathcal{O}

W naszym przykładzie alfabet

$\mathcal{A}_{O_I} = \{ 51223490121, 53490212634, 59912340978, 56767674510, 57601293847, 54345546900, \text{Informatyk_1}, \text{Prezes_1}, \text{Administracja_3} \}$.

Definicja 5.5 Ontologia $O[O_x, S]$

Ontologią $O[O_x, S]$ dla zbioru odwzorowań S nazywamy parę $\langle O_x, O_s[S] \rangle$ wyrażoną w alfabecie \mathcal{A}_O i składającą się z terminologii O_x oraz zbioru asercji $O_s[S]$ utworzonych na podstawie zbioru odwzorowań S , w taki sposób, że:

$$O_s[S] = \{ \{C(i): \exists \mathcal{M}_C \in \mathcal{S}_{\mathcal{M}_C} (i \in r_C)\} \cup \{R(i, j): \exists \mathcal{M}_R \in \mathcal{S}_{\mathcal{M}_R} (\langle i, j \rangle \in r_R)\} \cup \{A(i, v): \exists \mathcal{M}_A \in \mathcal{S}_{\mathcal{M}_A} (\langle i, v \rangle \in r_A)\} \}.$$

Przykład 5.1 cd. Ontologia $O[O_x, S]$

W naszym przykładzie ontologia $O[O_x, S] = \langle O_x, O_s[S] \rangle$ jest zdefiniowana następująco:
Terminologia O_x :

$$Kobieta \sqcup Mężczyzna \equiv Człowiek \quad (1)$$

$$Kobieta \sqcap Mężczyzna \equiv \perp \quad (2)$$

$$Pracownik \sqsubseteq Człowiek \quad (3)$$

$$Pracownik \sqsubseteq \exists \text{jestNaStanowisku}.Stanowisko \quad (4)$$

$$\exists \text{jestNaStanowisku}.\top \sqsubseteq Pracownik \quad (5)$$

$$\exists \text{jestNaStanowisku}.\neg \text{Stanowisko} \sqsubseteq \perp \quad (6)$$

$$Pracownik \sqsubseteq \exists \text{dostajeWynagrodzenie}.\langle 1000, 8000 \rangle \quad (7)$$

$$\exists \text{dostajeWynagrodzenie}.\langle 0, \infty \rangle \sqsubseteq Pracownik \quad (8)$$

$$Stanowisko \sqcap Człowiek \equiv \perp$$

Zbiór asercji O_s :

- (1) *Kobieta*(53490212634), (16) *jestNaStanowisku*(51223490121, Informatyk_1),
- (2) *Kobieta*(57601293847), (17) *jestNaStanowisku*(53490212634, Informatyk_1),
- (3) *Kobieta*(54345546900), (18) *jestNaStanowisku*(59912340978, Informatyk_1),
- (4) *Mężczyzna*(51223490121), (19) *jestNaStanowisku*(56767674510, Prezes_2),
- (5) *Mężczyzna*(59912340978), (20) *jestNaStanowisku*(57601293847, Administracja_3),
- (6) *Mężczyzna*(56767674510), (21) *jestNaStanowisku*(54345546900, Administracja_3),
- (7) *Pracownik*(51223490121), (22) *dostajeWynagrodzenie* (51223490121, 4000),
- (8) *Pracownik*(53490212634), (23) *dostajeWynagrodzenie* (53490212634, 4000),
- (9) *Pracownik*(59912340978), (24) *dostajeWynagrodzenie* (59912340978, 4000),
- (10) *Pracownik*(56767674510), (25) *dostajeWynagrodzenie* (56767674510, 7500),
- (11) *Pracownik*(57601293847), (26) *dostajeWynagrodzenie* (57601293847, 1500),
- (12) *Pracownik*(54345546900), (27) *dostajeWynagrodzenie* (54345546900, 1500),
- (13) *Stanowisko*(Informatyk_1),
- (14) *Stanowisko*(Prezes_2),
- (15) *Stanowisko*(Administracja_3).

Dla ontologii O zdefiniujemy poprawność i kompletność zbioru odwzorowań S . Zakładamy, że $\mathcal{K} = \langle O_x, O_s[S] \rangle$ jest bazą wiedzy dla ontologii O . Określmy zbiór zapytań Q zdefiniowany w zbiorze odwzorowań S :

$$Q = \{q_C: \exists \mathcal{M}_C \in \mathcal{S}_{\mathcal{M}_C} (\mathcal{M}_C = \langle C, q_C \rangle) \cup$$

$$\{q_R: \exists \mathcal{M}_R \in \mathcal{S}_{\mathcal{M}_R} (\mathcal{M}_R = \langle R, q_R \rangle) \cup$$

$$\{q_A: \exists \mathcal{M}_A \in \mathcal{S}_{\mathcal{M}_A} (\mathcal{M}_A = \langle A, q_A \rangle)\}.$$

Definicja 5.6 Poprawność zbioru odwzorowań S

Zbiór odwzorowań $S = S_{M_C} \cup S_{M_R} \cup S_{M_A}$ jest poprawny wtedy i tylko wtedy, gdy

- (1) dla wszystkich konceptów, dla których zostało zdefiniowane odwzorowanie, zapytanie q_C zwraca jedynie zbiór nazw osobników będących wystąpieniami konceptu C ; innymi słowy odpowiedź nie zawiera nazwy osobnika niebędącego wystąpieniem konceptu C :

$$\forall q_C \in Q (\neg \exists i ((i \in r_C) \wedge \neg (\mathcal{K} \models C(i))))$$
 - (2) dla wszystkich ról, dla których zostało zdefiniowane odwzorowanie, zapytanie q_R zwraca jedynie zbiór nazw osobników będących wystąpieniami roli R ; innymi słowy odpowiedź nie zawiera pary nazw osobników niebędącej wystąpieniem roli R :

$$\forall q_R \in Q (\neg \exists i, j ((\langle i, j \rangle \in r_R) \wedge \neg (\mathcal{K} \models R(i, j))) \vee ((\langle j, i \rangle \in r_R) \wedge \neg (\mathcal{K} \models R(j, i))))$$
 - (3) dla wszystkich atrybutów, dla których zostało zdefiniowane odwzorowanie zapytanie q_A zwraca jedynie zbiór par: nazwa osobnika, wartość, będących wystąpieniami atrybutu A ; innymi słowy odpowiedź nie zawiera pary: nazwa osobnika, wartość, niebędącej wystąpieniem atrybutu A :

$$\forall q_A \in Q (\neg \exists i ((\langle i, v \rangle \in r_A) \wedge \neg (\mathcal{K} \models A(i, v)))).$$
-

Definicja 5.7 Kompletność zbioru odwzorowań S

Zbiór odwzorowań $S = S_{M_C} \cup S_{M_R} \cup S_{M_A}$ jest kompletny wtedy i tylko wtedy, gdy

- (1) dla wszystkich konceptów, dla których zostało zdefiniowane odwzorowanie, zapytanie q_C zwraca zbiór wszystkich nazw osobników będących wystąpieniami konceptu C ; innymi słowy nie istnieje osobnik będący wystąpieniem konceptu C , którego nazwa nie jest zwracana zapytaniem q_C :

$$\forall q_C \in Q (\neg \exists i (\mathcal{K} \models C(i) \wedge i \notin r_C))$$
 - (2) dla wszystkich ról, dla których zostało zdefiniowane odwzorowanie, zapytanie q_R zwraca zbiór wszystkich par nazw osobników będących wystąpieniami roli R ; innymi słowy nie istnieje para osobników będąca wystąpieniem roli R , która to para nazw nie jest zwracana zapytaniem q_R :

$$\forall q_R \in Q (\neg \exists i, j ((\mathcal{K} \models R(i, j) \wedge \langle i, j \rangle \notin r_R) \vee (\mathcal{K} \models R(j, i) \wedge \langle j, i \rangle \notin r_R)))$$
 - (3) dla wszystkich atrybutów, dla których zostało zdefiniowane odwzorowanie, zapytanie q_A zwraca zbiór wszystkich par: nazwa osobnika, wartość, będących wystąpieniami atrybutu A ; innymi słowy nie istnieje para: nazwa osobnika, wartość, będąca wystąpieniem atrybutu A , która to para nie jest zwracana zapytaniem q_A :

$$\forall q_A \in Q (\neg \exists i (\mathcal{K} \models A(i, v) \wedge \langle i, v \rangle \notin r_A)).$$
-

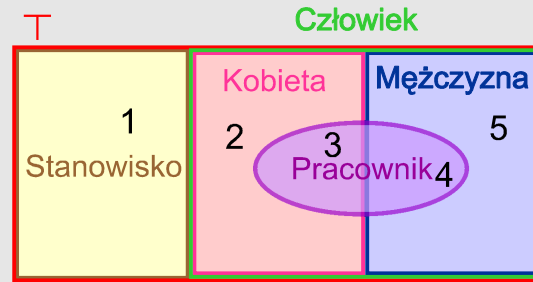
Przykład 5.1 cd. Poprawność zbioru odwzorowań S

W naszym przykładzie zbiór odwzorowań S jest poprawny. Następujące zdania są prawdziwe:

1. $\mathcal{K} \models Kobieta(53490212634), Kobieta(57601293847), Kobieta(54345546900);$
2. $\mathcal{K} \models Mężczyzna(51223490121), Mężczyzna(59912340978), Mężczyzna(56767674510);$

3. $\mathcal{K} \models \text{Pracownik}(51223490121), \text{Pracownik}(53490212634), \text{Pracownik}(59912340978),$
 $\text{Pracownik}(56767674510), \text{Pracownik}(57601293847), \text{Pracownik}(54345546900);$
4. $\mathcal{K} \models \text{Stanowisko}(\text{Informatyk_1}), \text{Stanowisko}(\text{Prezes_2}),$
 $\text{Stanowisko}(\text{Administracja_3});$
5. $\mathcal{K} \models \text{jestNaStanowisku}(51223490121, \text{Informatyk_1}),$
 $\text{jestNaStanowisku}(53490212634, \text{Informatyk_1}),$
 $\text{jestNaStanowisku}(59912340978, \text{Informatyk_1}),$
 $\text{jestNaStanowisku}(56767674510, \text{Prezes_2}),$
 $\text{jestNaStanowisku}(57601293847, \text{Administracja_3}),$
 $\text{jestNaStanowisku}(54345546900, \text{Administracja_3});$
6. $\mathcal{K} \models \text{dostajeWynagrodzenie}(51223490121, 4000),$
 $\text{dostajeWynagrodzenie}(53490212634, 4000),$
 $\text{dostajeWynagrodzenie}(59912340978, 4000),$
 $\text{dostajeWynagrodzenie}(56767674510, 7500),$
 $\text{dostajeWynagrodzenie}(57601293847, 1500),$
 $\text{dostajeWynagrodzenie}(54345546900, 1500);$

Wykażmy prawdziwość powyższych zdań, używając kartograficznej reprezentacji wiedzy. Tworzymy mapę konceptów dla przykładowej ontologii $\mathcal{O}[\mathcal{O}_x, S]$.



Rysunek 5.2. Mapa konceptów dla przykładowej ontologii

Rysunek 5.2 przedstawia mapę konceptów dla ontologii \mathcal{O} . Na mapie konceptów zaznaczono koncept *Pracownik*. Dodatkowo z ontologii \mathcal{O} wynikają poniższe zależności:

$\text{Pracownik} \equiv \exists \text{jestNaStanowisku}.\text{Stanowisko}$ oraz $\text{Pracownik} \equiv \exists \text{jestNaStanowisku}.\top$ (na podstawie (4), (5) i (6)),

$\text{Pracownik} \equiv \exists \text{dostajeWynagrodzenie}.\langle 1000, 8000 \rangle$ oraz

$\exists \text{dostajeWynagrodzenie}.(0, \infty)$ (na podstawie (7) i (8)).

Zdefiniujemy sygnatury s_C dla konceptów z ontologii \mathcal{O} :

$$s(\text{Kobieta}) = 01100$$

$$s(\text{Mężczyzna}) = 00011$$

$$s(\text{Człowiek}) = 01111$$

$$s(\text{Pracownik}) = 00110$$

$$s(\text{Stanowisko}) = 10000$$

$$s(\exists \text{jestNaStanowisku}.\text{Stanowisko}) = 00110$$

$$s(\exists \text{jestNaStanowisku}.\top) = 00110$$

$$s(\exists \text{jestNaStanowisku}.\neg \text{Stanowisko}) = 00000$$

$$s(\exists \text{dostajeWynagrodzenie}.\langle 1000, 8000 \rangle) = 00110$$

$$s(\exists \text{dostajeWynagrodzenie}.(0, \infty)) = 00110$$

Na podstawie zbioru asercji $\mathcal{O}_s[S]$ wyznaczamy sygnatury dla osobników zdefiniowanych w ontologii \mathcal{O} .

$s(53490212634) = s(Kobieta) \wedge s(Pracownik) = 01100 \wedge 00110 = 00100$
(z (1), (8), (17) i (23)),
 $s(57601293847) = s(Kobieta) \wedge s(Pracownik) = 01100 \wedge 00110 = 00100$
(z (2), (11), (20) i (26)),
 $s(54345546900) = s(Kobieta) \wedge s(Pracownik) = 01100 \wedge 00110 = 00100$
(z (3), (12), (21) i (27)),
 $s(51223490121) = s(Mężczyzna) \wedge s(Pracownik) = 00011 \wedge 00110 = 00010$
(z (4), (7), (16) i (22)),
 $s(51223490121) = s(Mężczyzna) \wedge s(Pracownik) = 00011 \wedge 00110 = 00010$
(z (5), (9), (18) i (24)),
 $s(56767674510) = s(Mężczyzna) \wedge s(Pracownik) = 00011 \wedge 00110 = 00010$
(z (6), (10), (19) i (25)),
 $s(\text{Informatyk}_1) = s(\text{Stanowisko}) = 10000$
(z (13), (16), (17) i (18)),
 $s(\text{Prezes}_2) = s(\text{Prezes}) = 10000$
(z (14) i (19)),
 $s(\text{Administracja}_3) = s(\text{Administracja}) = 10000$
(z (15), (20) i (21)).

Mając wyznaczone sygnatury osobników, możemy wyznaczyć wszystkie wystąpienia poszczególnych konceptów, korzystając z reguły $C(i) \Leftrightarrow s(i) \leq s(C)$.

Poniżej opisano zbiory wystąpień konceptów dla ontologii O :

Kobieta: 53490212634, 57601293847, 54345546900, ponieważ

$s(53490212634) \leq s(Kobieta)$, $s(57601293847) \leq s(Kobieta)$, $s(54345546900) \leq s(Kobieta)$ oraz dla żadnego innego osobnika nie zachodzi taka zależność. W analogiczny sposób możemy wyznaczyć zbiory wystąpień dla pozostałych konceptów. W przypadku ról i atrybutów wiadomo, że terminologia wymaga, aby każdy pracownik miał określoną wysokość wynagrodzenia oraz stanowisko. Dla każdego z pracowników te zależności są wprost podane w postaci asercji w $O_s[S]$. Dlatego żadne nowe wystąpienia ról i atrybutów nie mogą być wywnioskowane. W ten sposób dowiedliśmy prawdziwości zdań, a tym samym poprawności zbioru odwzorowań.

Poniżej pokażemy przykład, w którym zbiór odwzorowań jest niepoprawny. Załóżmy, że dla zapytania $q_{Pracownik}$ odpowiedź $r_{Pracownik}$ nie zawiera nazwy osobnika 53490212634:

$r_{Pracownik} = \{51223490121, 59912340978, 56767674510, 57601293847, 54345546900\}$.

Zbiór asercji $O_s[S]$ jest pomniejszony o asercję (8) *Pracownik*(53490212634). Jednak sygnatura osobnika 53490212634 nie uległa zmianie:

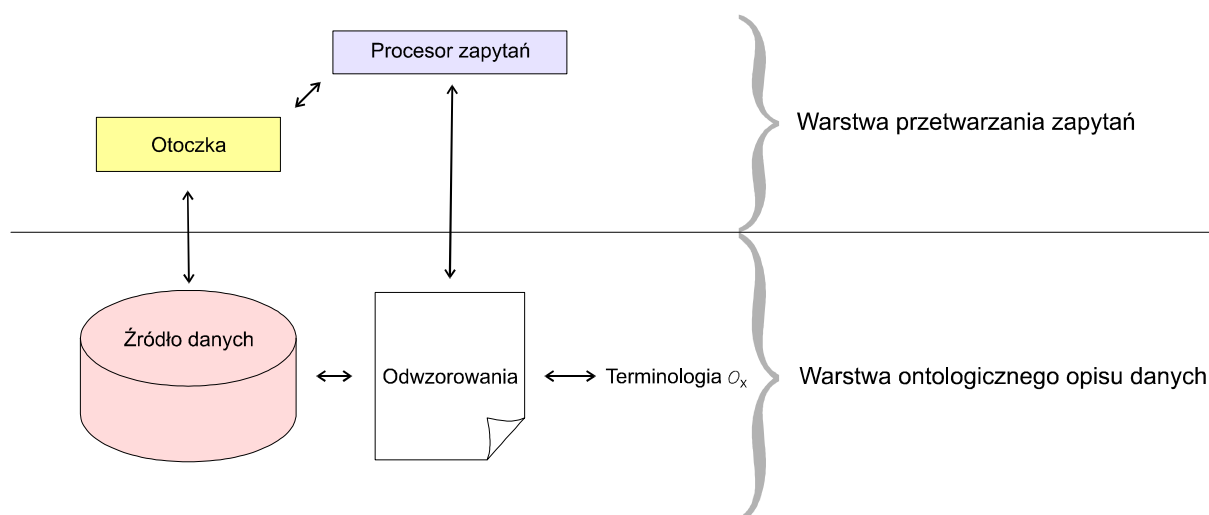
$s(53490212634) = s(Kobieta) \wedge s(Pracownik) = 01100 \wedge 00110 = 00100$
(z (1), (17) i (23)),

Zauważmy, że warunek 1 definicji 5.6 nie jest spełniony. Istnieje koncept *Pracownik*, dla którego zapytanie $q_{Pracownik}$ nie zwraca zbioru wszystkich osobników będących wystąpieniami konceptu *Pracownik*. Istnieje osobnik 53490212634 będący wystąpieniem konceptu *Pracownik*, ale nie należący do zbioru nazw osobników zwracanych przez zapytania $q_{Pracownik}$.

5.2 Budowanie źródła wiedzy dla danych opisanych ontologicznie

Metoda SED umożliwia ontologiczne opisanie danych w źródle danych w taki sposób, aby możliwe było utworzenie źródła wiedzy $\mathcal{KS} = \langle \mathcal{O}, \mathcal{Q}_O \rangle$, gdzie $\mathcal{O} = \langle \mathcal{O}_x, \mathcal{O}_s[S] \rangle$ jest ontologią opisaną w definicji 5.5, a S jest poprawnym zbiorem odwzorowań. Zbudowanie warstwy przetwarzania zapytań nad warstwą ontologicznego opisu danych umożliwia stworzenie algorytmów wnioskujących z danych zapisanych w zewnętrznym źródle danych – warstwy wiedzy (ang. *Knowledge Layer* – KL) [GOCZ2006b]. Metoda RED (*Reasoning over External Data*), korzystając z kartograficznej reprezentacji wiedzy, dostarcza zbioru algorytmów służących do odpowiadania na zapytania zdefiniowane w języku zapytań \mathcal{Q}_O .

W warstwie przetwarzania zapytań zostały zdefiniowane dwa komponenty: **procesor zapytań** i **otoczka**. Procesor zapytań jest odpowiedzialny za przeprowadzenie całego procesu wnioskowania z wykorzystaniem otoczki, do której są przekazywane zapytania zbudowane jako złożenia zapytań zdefiniowanych w zbiorze odwzorowań S . Głównym zadaniem metody RED jest wykorzystanie poprawnego i kompletnego zbioru odwzorowań S do wnioskowania o osobnikach z wykorzystaniem algorytmów operujących na kartograficznej reprezentacji wiedzy.



Rysunek 5.3. Wzbogacenie warstwy ontologicznego opisu danych o warstwę przetwarzania zapytań

Zadanie to wymaga uzyskania informacji o sygnaturach konceptów ze współpracującego z procesorem zapytań systemu KaSeA. Każde odwzorowanie konceptu przetwarzane jest na **odwzorowanie sygnatury**. Zapytanie pozostaje niezmienione, natomiast sygnatura konceptu wyliczana jest na podstawie sygnatur przechowywanych w systemie KaSeA.

Definicja 5.8 Odwzorowanie sygnatury

Dla każdego odwzorowania konceptu $\mathcal{M}_C = \langle C, q_C \rangle$ istnieje odwzorowanie sygnatury $\mathcal{M}_{s(C)}$ będące parą $\langle s(C), q_C \rangle$, gdzie $s(C)$ jest sygnaturą konceptu C .

Ze zbioru odwzorowań konceptów jest tworzony zbiór odwzorowań sygnatur $\mathcal{S}_{\mathcal{M}_{s(C)}}$.

Definicja 5.9 Zbiór odwzorowań sygnatur

Zbiór odwzorowań sygnatur jest zdefiniowany następująco:

$$\mathcal{S}_{\mathcal{M}_{s(C)}} = \{\mathcal{M}_{s(C)} = \langle s(C), q_C \rangle : \exists \mathcal{M}_C \in \mathcal{S}_{\mathcal{M}_C} (\mathcal{M}_C = \langle C, q_C \rangle)\}.$$

Język zapytań \mathcal{Q}_O umożliwia formułowanie zapytań o osobniki, w których to zapytaniach mogą występować koncepty złożone, wyrażone w logice opisowej \mathcal{L}_O na alfabecie \mathcal{A}_O . Pierwszym zadaniem metody RED jest wyznaczenie *sygnatury maksymalnego pokrycia* dla konceptu C zdefiniowanego w zapytaniu względem zbioru odwzorowań sygnatur. Zanim zdefiniujemy sygnaturę maksymalnego pokrycia posłużmy się przykładem.

Przykład 5.3 Sygnatura maksymalnego pokrycia

W naszym przykładzie usuńmy ze zbioru $\mathcal{S}_{\mathcal{M}_C}$ odwzorowanie $\mathcal{M}_{Kobieta}$:

$\mathcal{S}_{\mathcal{M}_C} = \{\mathcal{M}_{Mężczyzna}, \mathcal{M}_{Pracownik}, \mathcal{M}_{Stanowisko}\}$. Zbiór odwzorowań \mathcal{S} nadal pozostaje poprawny.

Wyobraźmy sobie, że w zapytaniu występuje koncept $Człowiek$. Dla sygnatury $s(Człowiek)$ nie jest zdefiniowane odwzorowanie. Wiemy jednak, że koncepty $Pracownik$ i $Mężczyzna$ dziedziczą od konceptu $Człowiek$. Analizując mapę konceptów dla ontologii O z rysunku 5.2, zauważymy, że obszary odpowiadające konceptom $Pracownik$ i $Mężczyzna$ w pełni nie pokrywają obszaru konceptu $Człowiek$. Z tych konceptów, dla których zostało zdefiniowane odwzorowanie (w naszym przypadku konceptów $Mężczyzna$, $Pracownik$ i $Stanowisko$ – pamiętajmy, że usunęliśmy odwzorowanie dla konceptu $Kobieta$), musimy stworzyć pewien koncept złożony, który pokrywa maksymalny (mierzony liczbą regionów) obszar konceptu $Człowiek$. Dla naszej przykładowej ontologii jest to koncept $Mężczyzna \sqcup Pracownik$. Poprzez sygnaturę maksymalnego pokrycia rozumiemy sygnaturę tego konceptu.

Definicja 5.10 Sygnatura maksymalnego pokrycia

Zdefiniujmy zbiór sygnatur \mathcal{S}_S dla zbioru odwzorowań sygnatur $\mathcal{S}_{\mathcal{M}_{s(C)}}$:

$$\mathcal{S}_S = \{s_g : \exists \mathcal{M}_{s(C)} \in \mathcal{S}_{\mathcal{M}_{s(C)}} (s_g = s(C))\}.$$

Sygnatura maksymalnego pokrycia s_{\max_C} dla konceptu C jest to sygnatura, która spełnia następujące warunki:

- 1) $s_{\max_C} \leq s(C)$,
 - 2) s_{\max_C} można zbudować jako sumę dowolnych iloczynów sygnatur ze zbioru \mathcal{S}_S ,
 - 3) nie istnieje sygnatura $s_g \leq s(C)$ taka, która jest sumą dowolnych iloczynów sygnatur ze zbioru \mathcal{S}_S i dla której $r(s_g) > r(s_{\max_C})$, gdzie $r(s)$ jest liczbą jedynek w sygnaturze s .
-

Twierdzenie 5.1 Twierdzenie o sygnaturach osobników

Sygnatura $s(i)$ dowolnego osobnika zdefiniowanego w ontologii $O = \langle O_x, O_s[S] \rangle$ jest równa iloczynowi pewnych sygnatur ze zbioru \mathcal{S}_S .

Dowód

Dla dowolnego osobnika i niech $S_C = \{C_1, \dots, C_n\}$ oznacza zbiór konceptów, dla których istnieje odwzorowanie $\mathcal{M}_C = \langle C, q_C \rangle$, gdzie i należy do zbioru nazw osobników będących odpowiedziami na zapytanie zdefiniowane w takim odwzorowaniu:

$$S_C = \{C: \exists \mathcal{M}_C \in \mathcal{S}_{\mathcal{M}_C} (\mathcal{M}_C = \langle C, q_C \rangle \wedge i \in r_C)\}.$$

Poprawność zbioru odwzorowań pozwala stworzyć zbiór asercji typu $C_i(i)$ dla każdego konceptu C ze zbioru S_C . Na podstawie tych asercji jest wyznaczana sygnatura osobnika i – zgodnie z założeniami metody kartograficznej. Jeżeli $S_i = \{s(C): C \in S_C\}$ oznacza zbiór sygnatur konceptów należących do zbioru S_C , to $s(i) = \bigcap_{s_k \in S_i} s_k$, gdzie $k=1, \dots, |S_i|$.

Twierdzenie 5.2 Twierdzenie o zbiorze wystąpień osobników konceptu C

Dla ontologii $\mathcal{O} = \langle \mathcal{O}_x, \mathcal{O}_s[S] \rangle$ zbiór wystąpień osobników konceptu C jest zbiorem osobników i , których sygnatura $s(i) \leq s_{\max_C}$.

Dowód

Z zasad metody kartograficznej wiemy, że $C(i) \Leftrightarrow s(i) \leq s(C)$. Aby twierdzenie 5.2 nie było prawdziwe, musi istnieć osobnik i , którego sygnatura $s(i) \leq s(C)$ i $s(i) \not\leq s_{\max_C}$. Ponieważ $s(i) = \bigcap_{s_k \in S_i} s_k$, gdzie

$$k=1, \dots, |S_i|, \text{ to istnieje } s_g = \bigcap_{s_k \in S_i} s_k \vee s_{\max_C}, \text{ gdzie } k=1, \dots, |S_i| \text{ i } s_g \leq s(C).$$

Ponieważ $r(s(i)) \neq 0$ (w przeciwnym razie baza wiedzy byłaby niespójna) i $s(i) \not\leq s_{\max_C}$, to $r(s_g) \geq r(s_{\max_C}) + 1$. Oznacza to, że s_{\max_C} nie jest sygnaturą maksymalnego pokrycia dla konceptu C (nie jest spełniony warunek 3 definicji 5.9), co przeczy założeniu.

Pierwszym wspomnianym zadaniem metody RED jest wyznaczenie sygnatury maksymalnego pokrycia dla zadanego konceptu C , czyli s_{\max_C} . Drugim zadaniem metody RED jest sformułowanie dla sygnatury s_{\max_C} zapytania do otoczki, które umożliwi pobranie wszystkich nazw osobników i o sygnaturach $s(i) \leq s_{\max_C}$.

Poniżej sformułujemy algorytm maksymalnego pokrycia MCA (*Maximum Coverage Algorithm*), który dla sygnatury zadanego konceptu C wyznacza sygnaturę s_{\max_C} i wyszukuje sumę iloczynów sygnatur z S_s , na podstawie których możliwe jest sformułowanie zapytania do otoczki zwracającego wszystkie nazwy osobników i o sygnaturach $s(i) \leq s_{\max_C}$. Algorytm MCA jest oparty na algorytmie Apriori [WITT2000].

Algorytm 5.1 Algorytm maksymalnego pokrycia

Wejście: Sygnatura s

Wyjście: Suma iloczynów sygnatur z S_s , na podstawie których możliwe jest sformułowanie zapytania zwracającego wszystkie nazwy osobników i o sygnaturach $s(i) \leq s_{\max_C}$.

1. **Jeżeli** sygnatura $s \in S_s$ (ma swoje odwzorowanie):
2. Zwróć sygnaturę s .
3. **W przeciwnym wypadku:**
4. Znajdź wszystkie sygnatury $s_g \in S_s$, które nie są rozłączne z s i nie są podrzędne w stosunku do s , a następnie zapisz je na listę l_0 .
5. Znajdź wszystkie sygnatury atomowe, które są podrzędne w stosunku do s , a następnie zapisz je na listę l_1 .
6. **Jeżeli** lista l_0 nie jest pusta:
7. Z listy l_1 usuń te sygnatury, które są podrzędne w stosunku do dowolnej innej sygnatury na tej liście.
8. **Dla każdej pary** (s_i, s_j) takich, że $s_i, s_j \in l_0$; $s_i \neq s_j$:
9. Oblicz sygnaturę $s_t = s_i \wedge s_j$ (zapamiętaj użyte sygnatury).
10. **Jeśli** s_t jest podrzędne w stosunku do s , dodaj s_t do listy l_1 .
11. **W przeciwnym wypadku** dodaj s_t do listy l_2 .
12. **Koniec** {Dla każdego}
13. **Dopóki** lista l_2 nie jest pusta:
14. Z listy l_1 usuń te sygnatury, które są podrzędne w stosunku do dowolnej innej sygnatury na tej liście.
15. Przenieś zawartość listy l_2 na listę l_3 i wyczyść listę l_2 .
16. **Dla każdej pary** (s_i, s_j) takich, że $s_i \in l_3$, $s_j \in l_0$:
17. Oblicz sygnaturę $s_t = s_i \wedge s_j$ (zapamiętaj użyte sygnatury).
18. **Jeśli** s_t jest podrzędne w stosunku do s , dodaj s_t do listy l_1 .
19. **W przeciwnym wypadku** dodaj s_t do listy l_2 .
20. **Koniec** {Dla każdego}
21. **Koniec** {Dopóki}
22. **Koniec** {Jeżeli}
23. Z listy l_1 usuń te sygnatury, które są podrzędne w stosunku do dowolnej innej sygnatury na tej liście.
24. Lista l_1 zawiera listę list sygnatur:
 $l_1 = ((s_{11}, s_{12}, \dots, s_{1n}), (s_{21}, s_{22}, \dots, s_{2m}), \dots, (s_{l1}, s_{l2}, \dots, s_{lp}))$.
 Zwróć tę listę jako sumę $S_1 \vee S_2 \vee \dots \vee S_l$, gdzie $S_1 = s_{11} \vee s_{12} \vee \dots \vee s_{1n}$,
 $S_2 = s_{21} \vee s_{22} \vee \dots \vee s_{2m}$, ..., $S_l = s_{l1} \vee s_{l2} \vee \dots \vee s_{lp}$.
25. **Koniec** {Jeżeli}

Innymi słowy, algorytm MCA zwraca wyrażenie składające się z l sum takich, że:

- 1) suma S_1 składa się z największych sygnatur $s_g \leq s(C)$ z S_s :
 $S_1 = s_{11} \vee s_{12} \vee \dots \vee s_{1n}$, gdzie $s_{11}, \dots, s_{1n} \leq s(C) \wedge \neg \exists s_q, s_r \in \{s_{11}, \dots, s_{1n}\} (s_q \leq s_r)$
- 2) suma S_2 składa się z największych par typu $s_i \wedge s_{i+1} \leq s(C)$ takich, że $s_i, s_{i+1} \in S_s$ oraz $s_i \wedge s_{i+1}$ nie jest podrzędna względem żadnej sygnatury z S_1
 $S_2 = s_{21} \vee s_{22} \vee \dots \vee s_{2m}$, gdzie
 $s_{21}, \dots, s_{2m} \leq s(C) \wedge \forall s_q \in \{s_{21}, \dots, s_{2m}\} ((s_q = s_i \wedge s_{i+1}) \wedge (s_i, s_{i+1} \in S_s) \wedge (\neg \exists s_r \in \{s_{11}, \dots, s_{1n}\} (s_q \leq s_r)))$

3) suma S_3 składa się z największych trójek typu $s_i \wedge s_{i+1} \wedge s_{i+2} \leq s(C)$ takich, że $s_i, s_{i+1}, s_{i+2} \in S_s$ oraz $s_i \wedge s_{i+1} \wedge s_{i+2}$ nie jest podrzędna względem żadnej sygnatury z S_1 i żadnego przecięcia sygnatur z S_2

$S_3 = s_{31} \vee s_{32} \vee \dots \vee s_{3k}$, gdzie

$$s_{31}, \dots, s_{3k} \leq s(C) \wedge \forall s_q \in \{s_{31}, \dots, s_{3k}\} ((s_q = s_i \wedge s_{i+1} \wedge s_{i+2}) \wedge (s_i, s_{i+1}, s_{i+2} \in S_s) \wedge (\neg \exists s_r \in \{s_{11}, \dots, s_{1n}, s_{21}, \dots, s_{2m}\} (s_q \leq s_r)))$$

...

l) suma S_l składa się z największych l -ek typu $s_i \wedge \dots \wedge s_{i+l-1} \leq s(C)$ takich, że $s_i, \dots, s_{i+l-1} \in S_s$ oraz $s_i \wedge \dots \wedge s_{i+l-1}$ nie jest podrzędna względem żadnej sygnatury z S_1 i żadnego przecięcia sygnatur z S_2, \dots, S_{l-1} :

$S_l = s_{l1} \vee s_{l2} \vee \dots \vee s_{lo}$, gdzie

$$s_{l1}, \dots, s_{lo} \leq s(C) \wedge \forall s_q \in \{s_{l1}, \dots, s_{lo}\} ((s_q = s_i \wedge \dots \wedge s_{i+l-1}) \wedge (s_i, \dots, s_{i+l-1} \in S_s) \wedge (\neg \exists s_r \in \{s_{11}, \dots, s_{1n}, s_{21}, \dots, s_{2m}, \dots, s_{l-11}, \dots, s_{l-1s}\} (s_q \leq s_r)))$$

Twierdzenie 5.3 Twierdzenie o wartości sygnatury maksymalnego pokrycia

Sygnatura maksymalnego pokrycia $s_{\max_C} = S_1 \vee S_2 \vee \dots \vee S_l$.

Dowód

Aby $S_1 \vee S_2 \vee \dots \vee S_l$ było sygnaturą maksymalnego pokrycia, musi spełniać następujące warunki:

1) $S_1 \vee S_2 \vee \dots \vee S_l \leq s(C)$.

Warunek 1 jest spełniony, ponieważ

$$\forall s_i, \dots, s_j \leq s(C) \Rightarrow s_i \vee \dots \vee s_j \leq s(C).$$

2) $S_1 \vee S_2 \vee \dots \vee S_l$ można zbudować jako sumę dowolnych iloczynów sygnatur ze zbioru S_s .

Warunek 2 jest spełniony, ponieważ $S_1 \vee S_2 \vee \dots \vee S_l$ jest zdefiniowane jako suma iloczynów sygnatur ze zbioru S_s .

3) Nie istnieje sygnatura $s_g \leq s(C)$ taka, która jest sumą dowolnych iloczynów sygnatur ze zbioru S_s i dla której

$$r(s_g) > r(S_1 \vee S_2 \vee \dots \vee S_l), \text{ gdzie } r(s) \text{ jest liczbą jedynek}$$

w sygnaturze s .

Jeśli s_g jest sumą dowolnych iloczynów sygnatur ze zbioru S_s taką, że $s_g \leq s(C)$, to:

- jeżeli s_g jest sumą iloczynów będących jedną z sum w sygnaturze $S_1 \vee S_2 \vee \dots \vee S_l$, to $r(s_g) \leq r(S_1 \vee S_2 \vee \dots \vee S_l)$,

- jeżeli s_g nie jest sumą iloczynów będących jedną z sum w sygnaturze $S_1 \vee S_2 \vee \dots \vee S_l$, to istnieje sygnatura $s_i \geq s_g$ lub suma sygnatur $s_j \vee \dots \vee s_j \geq s_g$ będące elementami sygnatury $S_1 \vee S_2 \vee \dots \vee S_l$. Stąd $r(s_g) \leq r(S_1 \vee S_2 \vee \dots \vee S_l)$,

co dowodzi warunku 3.

W kolejnym kroku dowiedzimy, że na podstawie wyrażenia zwróconego przez algorytm MCA możliwe jest zbudowanie zapytania, które zwróci wszystkie nazwy osobników i , których sygnatury $s(i) \leq s_{\max_C}$. Na podstawie wyrażenia określającego sumę przecięć sygnatur $S_1 \vee S_2 \vee \dots \vee S_l$ budowane jest zapytanie q . Zapytanie q budowane jest w taki sposób, że:

- każda sygnatura w wyrażeniu jest wymieniana na odpowiadające jej zapytanie zdefiniowane w odwzorowaniu sygnatury,
- operator sumy przekłada się na sumę wyników poszczególnych zapytań,
- operator przecięcia przekłada się na iloczyn wyników poszczególnych zapytań.

Na mocy twierdzenia 5.1 wiemy, że sygnatura $s(i)$ dowolnego osobnika zdefiniowanego w ontologii $\mathcal{O} = \langle \mathcal{O}_x, \mathcal{O}_s[S] \rangle$ jest równa iloczynowi pewnych sygnatur ze zbioru S_s , a dokładnie:

$$s(i) = \bigcap_{s_k \in S_i} s_k, \text{ gdzie } k=1, \dots, |S_i| \text{ i } S_i = \{s(C): \exists \mathcal{M}_{s(C)} \in \mathcal{S}_{\mathcal{M}_{s(C)}} (\mathcal{M}_{s(C)} = \langle s(C), q_C \rangle \wedge i \in r_C)\}.$$

Założmy, że istnieje osobnik i o sygnaturze $s(i) \leq s_{\max_C}$, który nie został zwrócony przez zapytanie utworzone na podstawie wyrażenia sygnaturowego obliczonego algorytmem MCA, czyli $S_1 \vee S_2 \vee \dots \vee S_l$. Oznacza to, że sygnatura $s(i)$ nie jest podrzędna względem żadnej z sygnatur w sumie S_1 , ani względem żadnego przecięcia sygnatur z sum S_2, \dots, S_l , co jest równoważne stwierdzeniu, że $s(i)$ nie ma części wspólnej z s_{\max_C} , co przeczy założeniu.

Po zdefiniowaniu algorytmu MCA możliwe jest opisanie algorytmów używanych podczas wyszukiwania odpowiedzi na zapytania o osobniki, zdefiniowane w języku zapytań $Q_{\mathcal{O}}$.

Algorytm 5.2 Algorytm określania zbioru wystąpień conceptów

Wejście: Koncept C .

Wyjście: Zbiór wystąpień conceptu C .

1. Znajdź sygnaturę conceptu C .
2. Znajdź zapytanie dla tej sygnatury:
 - 2.a. Jeśli nie istnieje w bazie danych odwzorowanie sygnatury w zapytanie:
 - 2.b. Znajdź najbardziej odpowiednie zapytanie, używając algorytmu MCA.
 - 2.c. W przeciwnym wypadku:
 - 2.d. Pobierz zapytanie ze zbioru odwzorowań sygnatur.
3. Wykonaj zapytanie.
4. Zwróć wynik zapytania.

Algorytm 5.3 Algorytm sprawdzania przynależności

Wejście: Koncept C , nazwa osobnika i .

Wyjście: *True*, jeżeli i jest wystąpieniem conceptu C , *false*, jeżeli i jest wystąpieniem conceptu $\neg C$, lub *maybe*, jeżeli i nie jest wystąpieniem ani conceptu C , ani $\neg C$.

1. Znajdź sygnaturę conceptu C
2. Znajdź zapytanie dla tej sygnatury:
 - 2.a. Jeśli nie istnieje w bazie danych odwzorowanie sygnatury w zapytanie:
 - 2.b. Znajdź najbardziej odpowiednie zapytanie, używając algorytmu MCA.
 - 2.c. W przeciwnym wypadku:
 - 2.d. Pobierz zapytanie ze zbioru odwzorowań sygnatur.
3. Wykonaj zapytanie.
8. Sprawdź, czy osobnik i znajduje się wśród wyników zapytania.
9. Jeśli osobnik i znajduje się wśród wyników zapytania:
10. Zwróć *true*.

11. W przeciwnym wypadku:
12. Znajdź sygnaturę dla konceptu $\neg C$.
13. Znajdź zapytanie dla tej sygnatury:
 - 13.a. Jeśli nie istnieje w bazie danych odwzorowanie sygnatury w zapytanie:
 - 13.b. Znajdź najbardziej odpowiednie zapytanie, używając algorytmu MCA.
 - 13.c. W przeciwnym wypadku:
 - 13.d. Pobierz zapytanie ze zbioru odwzorowań sygnatur.
14. Wykonaj zapytanie.
15. Sprawdź, czy osobnik i znajduje się wśród wyników zapytania.
16. Jeśli osobnik i znajduje się wśród wyników zapytania:
17. Zwróć *false*.
18. W przeciwnym wypadku:
19. Zwróć *maybe*.

5.3 Podsumowanie

Mimo iż zewnętrzne źródła danych modelowane są zgodnie z założeniem o świecie zamkniętym CWA (*Closed World Assumption*) [REIT78], metody SED i RED otwierają ten świat i pozwalają na uzyskanie odpowiedzi zgodnych z założeniem o świecie otwartym OWA (*Open World Assumption*). Wynika to przede wszystkim stąd, że metoda SED umożliwia stworzenie ontologii, składającej się z terminologii i zbioru asercji, na podstawie danych przechowywanych w zewnętrznych źródłach. Dlatego algorytmy wnioskowania zdefiniowane w metodzie RED przetwarzają w rzeczywistości dane ontologiczne (nawet jeżeli są one przechowywane jako dane relacyjne). Dane ontologiczne zaś, z definicji, są danymi ze świata otwartego, a więc takimi, które postrzegamy jako niekompletne.

Poniżej przedstawiono inne najważniejsze cechy obu metod:

- 1) Metoda RED jest niezależna od formatu źródła danych – dla strukturalnych źródeł danych takich jak źródła XML, relacyjne bazy danych czy jakiegokolwiek inne źródła, dla których istnieje język zapytań oraz procesor tych zapytań, jedynym wymaganiem jest utworzenie metodą SED zbioru poprawnych odwzorowań.
- 2) Metoda SED jest zależna od formatu źródła danych tylko w zakresie definiowania zapytań w odwzorowaniach.
- 3) Obie metody mogą zostać użyte dla źródeł danych, dla których nie zdefiniowano języka zapytań (przykładowo pliki XLS) – należy wówczas opracować język zapytań (potencjalnie bardzo prosty) i zbudować odpowiednią otoczkę.
- 4) Wszystkie zmiany w źródle danych, które nie wpływają na strukturę samego źródła, są zawsze widoczne w odpowiedziach; nie są wymagane żadne aktualizacje bazy wiedzy.
- 5) Zmiana struktury źródła danych wymaga zdefiniowania nowych odwzorowań.
- 6) Nie są wymagane żadne zaawansowane techniki aktualizacji bazy wiedzy.

W tym rozdziale został spełniony cel drugi rozprawy: opracowanie algorytmów i metod, służących do wytworzenia zamodelowanego systemu. W rozdziale tym opisano algorytmy i metody tworzenia źródła wiedzy dla zewnętrznego źródła danych. W rozdziale następnym – szóstym opisano algorytmy i metody służące do przetwarzania zapytań w systemie integracji wiedzy.

Rozdział 6. Metoda przetwarzania zapytań w środowisku rozproszonego systemu integracji wiedzy

6.1 Przetwarzanie zapytań w środowisku rozproszonym

W rozdziale 3 przedstawiono model systemu integracji wiedzy. Następnie w rozdziale 4 i 5 przedstawiono dwa rodzaje źródeł wiedzy, które spełniają wymagania stawiane przez system KISS integrowanym źródłom wiedzy. W niniejszym rozdziale opisano:

- 1) sposób przechowywania zależności pomiędzy terminologiami w systemie integracji wiedzy oraz
- 2) algorytmy przetwarzania zapytań w systemie integracji wiedzy.

Model systemu KISS zakłada, że w każdym węźle poprawnej hierarchii terminologii znajduje się pewna terminologia. Pojęcia z tej terminologii przekładają się na pewne pojęcia z terminologii importowanych. Podrozdział 6.2 rozdziału przedstawia sposób opisywania takich zależności między pojęciami z dwóch terminologii. Zależności te są niezbędne w celu odpowiedniego przetworzenia zapytania zadanego do danego węzła w hierarchii terminologii. W algorytmie SPIDER przedstawionym w podrozdziale 6.3 sprecyzowano, w jaki sposób zależności między pojęciami są wykorzystywane do znajdowania odpowiedzi na zapytania. Algorytm SPIDER dostosowuje algorytmy wnioskujące zaproponowane w metodzie kartograficznej do sytuacji, w której baza wiedzy jest rozproszona.

6.2 Zależności pomiędzy terminologiami w systemie integracji wiedzy

W rozdziale 3 pokazano, w jaki sposób problem łączenia terminologii, przy założeniu definicji zbioru odwzorowań aksjomatycznych, został sprowadzony do problemu zarządzania ontologiami powiązanymi między sobą relacją importu. W tym podrozdziale zostanie przedstawiona metoda przechowywania zależności między terminologiami z wykorzystaniem kartograficznej reprezentacji wiedzy.

W algorytmie 6.1 pokazano metodę budowania zależności między pojęciami zdefiniowanymi w terminologiach z poprawnej hierarchii terminologii. Ogólna idea algorytmu polega na tym, że dla każdej krawędzi w poprawnej hierarchii terminologii jest budowana *pajęczyna* powiązań między pojęciami zdefiniowanymi w dwóch terminologiach. Proces budowania pajęczyn jest procesem przygotowawczym, który jest wykonywany jednokrotnie po utworzeniu poprawnej hierarchii terminologii. Powtórzenie tego procesu jest

wymagane jedynie przy zmianie terminologii. Dopiero po utworzeniu poprawnych zależności między pojęciami zdefiniowanymi w różnych terminologiach możliwe jest wykonywanie zapytań do systemu KISS.

Algorytm 6.1 Budowanie zależności między terminologiami w hierarchii terminologii \mathcal{H}_T

Wejście: Poprawna hierarchia terminologii \mathcal{H}_T

Wyjście: Poprawna hierarchia terminologii wzbogacona o zależności między terminologiami

1. Stwórz mapę konceptów dla każdej terminologii O_x ze zbioru S_O .
2. Dla każdej krawędzi e_{ij} z wierzchołka v_i do v_j stwórz zbiór zależności między terminologią nadrzędną $O_{x_j} = f(v_j)$ a terminologią podrzędną $O_{x_i} = f(v_i)$.

Zależności między terminologią nadrzędną O_{x_j} a terminologią podrzędną O_{x_i} są reprezentowane w postaci *pajęczyny*. Dla uproszczenia w poniższej definicji przyjęto zapis $I = MD(O_x)$ oznaczający, że I jest modelem terminologii O_x .

Definicja 6.1 Pajęczyna $P(O_{x_n}, O_{x_p})$

Pajęczyną $P: O_{x_n} \rightarrow O_{x_p}$ pomiędzy terminologią nadrzędną O_{x_n} a terminologią podrzędną O_{x_p} nazywamy przyporządkowanie $p(o_n) = o_p$ przyporządkowujące obszarom z mapy konceptów terminologii nadrzędnej O_{x_n} obszary z mapy konceptów terminologii podrzędnej O_{x_p} .

Przyporządkowanie p musi spełniać następujący warunek:

$$\forall (I = MD(O_{x_n}) \wedge I = MD(O_{x_p})) (C(o_p)^I \subseteq C(o_n)^I),$$

gdzie $C(o)$ oznacza dowolny koncept odpowiadający obszarowi o .

Przykład 6.1

Przyjmijmy pewną hierarchię terminologii \mathcal{H}_T składającą się z trzech terminologii $S_O = \{A, B, C\}$, przy czym terminologia C importuje terminologie A i B .

Terminologia A :

$$\text{Człowiek} \equiv \top$$

$$\text{Kobieta} \sqcap \text{Mężczyzna} \equiv \perp$$

$$\text{Kobieta} \sqcup \text{Mężczyzna} \equiv \top$$

Terminologia B :

$$\text{Człowiek} \equiv \top$$

$$\exists \text{maDziecko}.\top \sqsubseteq \text{Człowiek}$$

Terminologia C :

$$\text{Dziecko} \sqsubseteq \text{Człowiek}$$

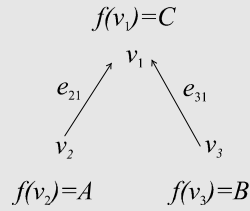
$$\exists \text{maDziecko}.\top \sqsubseteq \neg \text{Dziecko}$$

$$\text{Człowiek} \equiv \top \text{ (aksjomat zaimportowany z } A \text{ i } B)$$

$$\text{Kobieta} \sqcap \text{Mężczyzna} \equiv \perp \text{ (aksjomat zaimportowany z } A)$$

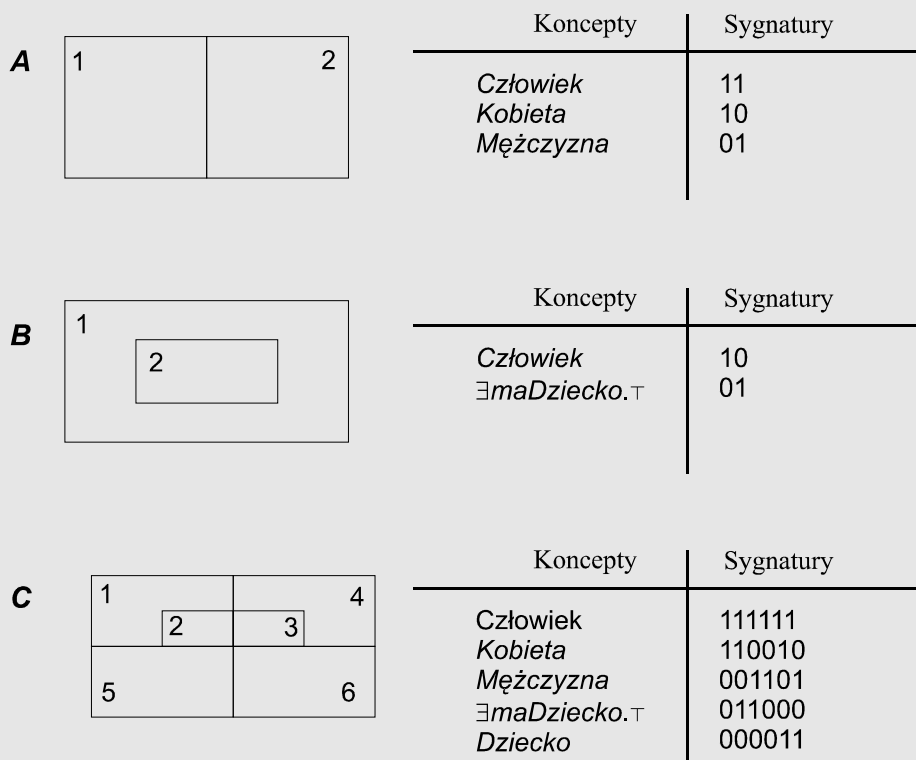
$Kobieta \sqcup Mężczyzna \equiv \top$ (aksjomat zaimportowany z A)
 $\exists maDziecko.\top \sqsubseteq Człowiek$ (aksjomat zaimportowany z B)

Rysunek 6.1 przedstawia hierarchię terminologii $\mathcal{H}_{\mathcal{T}}$.



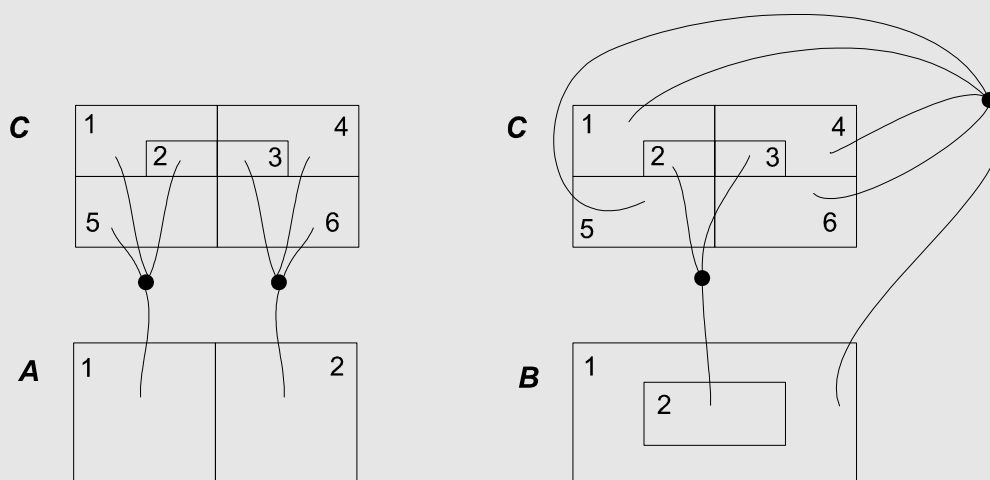
Rysunek 6.1. Przykładowa hierarchia terminologii

W grafie hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$ istnieją dwie krawędzie: e_{21} i e_{31} . Dla każdej krawędzi musi zostać stworzona odpowiednia pajęczyna opisująca zależności pomiędzy terminologią C i A oraz B i C . Zanim opiszemy te pajęczyny, pokażemy mapy konceptów dla poszczególnych terminologii. Rysunek 6.2 przedstawia odpowiednio mapy konceptów dla terminologii A , B i C . Na mapie konceptów terminologii A mamy dwa obszary atomowe. Obszar atomowy 1 odpowiada konceptowi *Kobieta*, zaś obszar atomowy 2 konceptowi *Mężczyzna*. Koncept *Człowiek* obejmuje oba obszary atomowe z tej mapy konceptów. Również na mapie konceptów terminologii B znajdują się jedynie dwa obszary atomowe. Obszar 1 odpowiada ludziom bezdzietnym ($\neg \exists maDziecko.\top$), zaś obszar drugi rodzicom ($\exists maDziecko.\top$). Mapa konceptów dla terminologii C składa się z sześciu obszarów atomowych. Zauważmy, że przecięcie obszarów z map konceptów terminologii A i B tworzy cztery obszary atomowe odpowiadające kobietom bezdzietnym, mężczyznom bezdzietnym, matkom i ojcom. Dodatkowo każdy z tych obszarów jest dzielony na dwa podobszary: dzieci i dorosłych. Aksjomat $\exists maDziecko.\top \sqsubseteq \neg Dziecko$ eliminuje jednak obszary dotyczące dzieci mających dzieci. Dlatego ostatecznie mapa konceptów terminologii C składa się jedynie z sześciu obszarów atomowych, a nie z ośmiu – dwa obszary odpowiadające matkom będącym dziećmi i ojcom będącym dziećmi są niespełnialne.



Rysunek 6.2. Hierarchia źródeł wiedzy w odniesieniu do hierarchii terminologii

Na rysunku 6.3 pokazane są pajęczyny pomiędzy terminologiami.



Rysunek 6.3. Pajęczyny pomiędzy terminologiami A i B oraz B i C

Pajęczyny obrazują, w jaki sposób obszary z terminologii nadrzędnych są odwzorowywane w obszary ontologii podrzędnych. Obszary 1, 2 i 5 terminologii C odpowiadają konceptowi *Kobieta*. Na mapie konceptów terminologii A konceptowi *Kobieta* odpowiada obszar atomowy 1. Dlatego też pajęczyna łączy ze sobą obszar 1 z terminologii A z obszarami 1, 2 i 3 z terminologii C. Analogiczna sytuacja zachodzi w przypadku konceptu *Mężczyzna*. Pajęczyna pomiędzy terminologią C i B buduje odwzorowania dla konceptów oznaczających ludzi bezdzietnych i rodziców. Ludziom bezdzietnym w terminologii C odpowiadają obszary atomowe 1, 4, 5 i 6, zaś w terminologii B obszar 1. Obszary 2 i 3 terminologii C odpowiadają obszarowi 2 terminologii B i odnoszą się do konceptu opisującego rodziców.

6.3 Dostosowanie algorytmów wnioskujących z wiedzy reprezentowanej kartograficznie do warunków jej rozproszenia

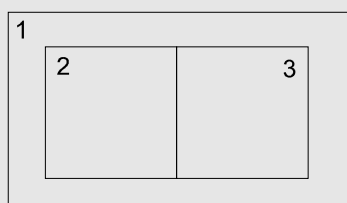
W procesie znajdowania odpowiedzi na zapytania dotyczące opisu świata w metodzie kartograficznej kluczowe jest wyznaczenie sygnatur osobników. Dostosowanie algorytmów wnioskujących z wiedzy reprezentowanej kartograficznie wymaga zdefiniowania metod wyznaczania sygnatur w środowisku rozproszonym. W systemie integracji wiedzy sygnatury osobników wyznaczane są w procesie znajdowania odpowiedzi na zapytania, inaczej niż w systemie KaSeA. Przesłanki przemawiające za wyznaczaniem sygnatur na tym etapie są właściwie analogiczne jak w przypadku warstwy wiedzy wnioskującej z danych zapisanych w zewnętrznym źródle danych. Wynika to z faktu, że różne źródła wiedzy w systemie integracji wiedzy mogą się zmieniać, tak jak i źródła zewnętrzne. Pożądane jest, aby system integracji wiedzy dostarczał informacji najbardziej aktualnych.

Przykład 6.2 Sygnatury osobników w systemie integracji wiedzy

Posłużmy się systemem KISS z przykładu 3.2. W celu analizy metod wyznaczania sygnatur osobników w środowisku rozproszonym przyjmijmy za przykładowe zapytanie $instances(Mężczyzna \sqcap Pracownik)(O_{x_2})$. Wyznaczamy sygnatury osobników w węźle v_2 w hierarchii terminologii. W tym celu wyznaczamy sygnatury osobników również w węzłach v_3 i v_4 .

Węzeł v_3 : $f(v_3) = O_{x_3}$

Mapa konceptów:



Koncepty	Sygnatury
Człowiek	011
Kobieta	010
Mężczyzna	001

\mathcal{KS}_3 :

$Mężczyzna(Linguini)$

Sygnatury osobników:

$s(Linguini) = 001$

v_3 :

$s(Linguini) = 001$

Węzeł v_4 : $f(v_4) = O_{x_4}$

Mapa konceptów:



Koncepty	Sygnatury
<i>Stanowisko</i>	010
<i>Pracownik</i>	001
$\exists \text{jestNaStanowisku.Stanowisko}$	001

\mathcal{KS}_4 :

$\text{jestNaStanowisku}(\text{Linguini}, \text{Kucharz})$

Sygnatury osobników:

$s(\text{Linguini}) = 001$ (dziedzina roli jestNaStanowisku jest koncept *Pracownik*),

$s(\text{Kucharz}) = 010$ (zakresem roli jestNaStanowisku jest koncept *Stanowisko*)

\mathcal{KS}_5

$\text{jestNaStanowisku}(\text{Kopciuszek}, \text{Sprzątaczką})$

Sygnatury osobników:

$s(\text{Kopciuszek}) = 001$,

$s(\text{Sprzątaczką}) = 010$

v_3 :

$s(\text{Linguini}) = 001$,

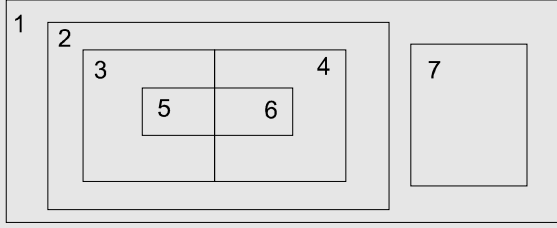
$s(\text{Kucharz}) = 010$,

$s(\text{Kopciuszek}) = 001$,

$s(\text{Sprzątaczką}) = 010$

Węzeł v_2 : $f(v_2) = O_{x_2}$

Mapa konceptów:



Koncepty	Sygnatury
Człowiek	0011110
Kobieta	0010100
Mężczyzna	0001010
Stanowisko	0000001
Pracownik	0000110
\exists jestNaStanowisku.Stanowisko	0000110
Ssak	0111110

\mathcal{KS}_2

$Ssak(\text{ChipWiewiórka})$

$Ssak(\text{DaleWiewiórka})$

$Pracownik(\text{Linguini})$

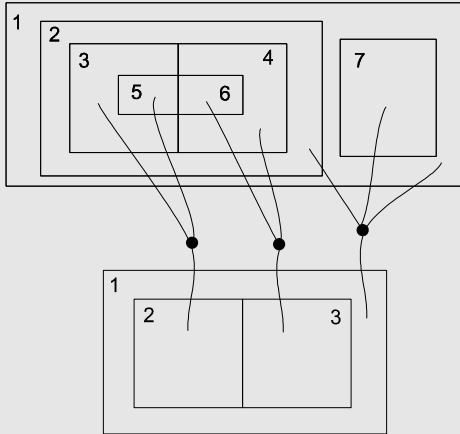
$s(\text{ChipWiewiórka}) = 0111110$

$s(\text{DaleWiewiórka}) = 0111110$

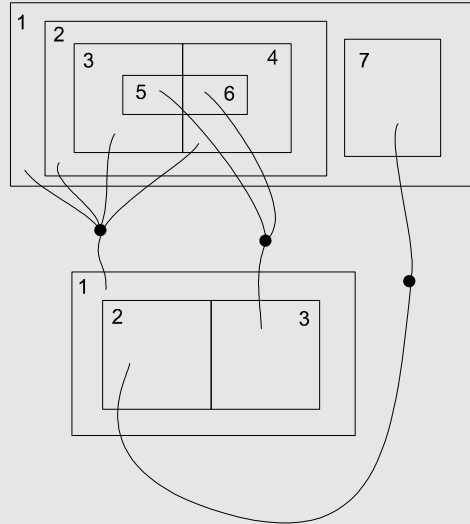
$s(\text{Linguini}) = 0000110$

Pajęczyny:

e_{32}



e_{42}



v_2 :

Nazwijmy pajęczynę $P(O_{x_3}, O_{x_2})$ dla krawędzi e_{32} odwzorowaniem p_1 oraz pajęczynę $P(O_{x_4}, O_{x_2})$ dla krawędzi e_{42} odwzorowaniem p_2 . Przyjmijmy, że o_s oznacza obszar o odpowiadający sygnaturze s , zaś s_{o_s} jest sygnaturą s dla obszaru o_s .

$$s(\text{Linguini}) = 0000110 \wedge s_{p_1(o_{001})} \wedge s_{p_2(o_{001})} = 0000110 \wedge s_{o_{0001010}} \wedge s_{o_{0000110}} = \\ 0000110 \wedge 0001010 \wedge 0000110 = 0000010$$

$$s(\text{Kucharz}) = s_{p_2(o_{010})} = s_{o_{0000001}} = 0000001$$

$$s(\text{Kopciuszek}) = s_{p_1(o_{001})} = 0010100,$$

$$s(\text{Sprzątaczką}) = s_{p_2(o_{010})} = 0000001$$

$$s(\text{ChipWiewiórka}) = 0111110$$

$$s(\text{DaleWiewiórka}) = 0111110$$

Przeliczanie sygnatur pokażemy na przykładzie osobnika *Linguini*. Ze źródła \mathcal{KS}_3 wiemy, że *Linguini* jest mężczyzną. Ponieważ źródło \mathcal{KS}_3 jest jedynym źródłem przypisanym do węzła v_3 , to cała wiedza o *Linguini* w tym węźle ogranicza się do wiedzy zawartej w źródle \mathcal{KS}_3 . Stąd sygnatura osobnika *Linguini* w węźle v_3 jest równa sygnaturze konceptu *Mężczyzna* na mapie konceptów terminologii O_{x_3} (001). Ponadto pajęczyna p_1 dostarcza informacji o odwzorowaniu sygnatury 001 z mapy konceptów dla terminologii O_{x_3} w sygnaturę z mapy konceptów terminologii O_{x_2} ($s_{p_1(o_{001})}$). Podobna sytuacja zachodzi dla węzła v_4 . Do węzła v_4 przypisane są dwa źródła wiedzy: \mathcal{KS}_4 i \mathcal{KS}_5 . Jednak w źródle \mathcal{KS}_5 nie ma żadnej informacji o osobniku *Linguini*. Sygnatura osobnika *Linguini* w węźle v_3 jest równa sygnaturze konceptu *Pracownik* na mapie konceptów terminologii O_{x_4} (001), która odpowiada sygnaturze 0000110 na mapie konceptów terminologii O_{x_2} (wynika to z odwzorowań zapisanych w pajęczynie p_2). Sygnatura osobnika *Linguini* w węźle v_2 musi odzwierciedlać wiedzę zebraną ze źródeł wiedzy $\mathcal{KS}_2, \mathcal{KS}_3, \mathcal{KS}_4, \mathcal{KS}_5$. Dlatego sygnatura osobnika *Linguini* w węźle v_2 jest przecięciem sygnatur 0000110 (odzwierciedlającą wiedzę zebraną ze źródła \mathcal{KS}_2) 0010100 (odzwierciedlającą wiedzę zebraną ze źródła \mathcal{KS}_3) i 0000110 (odzwierciedlającą wiedzę zebraną ze źródeł \mathcal{KS}_4 i \mathcal{KS}_5).

Odpowiadając na pytanie $instances(Mężczyzna \sqcap Pracownik)(O_{x_2})$ zgodnie z założeniami metody kartograficznej, wyznaczamy sygnaturę konceptu $Mężczyzna \sqcap Pracownik$ dla terminologii O_{x_2} równą $0001010 \wedge 0000110 = 0000010$. Odpowiedzią na to zapytanie jest zbiór osobników o sygnaturze s nie większej niż (\leq) sygnatura konceptu $Mężczyzna \sqcap Pracownik$, czyli 0000010. W omawianym przykładzie jest to zbiór jednoelementowy $\{Linguini\}$.

Definicja 6.2 Sygnatura nadrzędna $s_n[o, P(O_{x_n}, O_{x_p}), P_{\mathcal{KS}}]$ osobnika o dla pajęczyny $P(O_{x_n}, O_{x_p})$ dla zbioru źródeł wiedzy $P_{\mathcal{KS}}$

Sygnaturą nadrzędna s_n osobnika o dla pajęczyny $P(O_{x_n}, O_{x_p})$ nazywamy sygnaturę osobnika wyznaczoną dla terminologii nadrzędnej O_{x_n} i odpowiadającą sygnaturze osobnika o dla terminologii O_{x_p} dla źródeł wiedzy $P_{\mathcal{KS}}$.

Definicja 6.3 Sygnatura podrzędna $s_p[o, P(O_{x_n}, O_{x_p}), P_{\mathcal{KS}}]$ osobnika o dla pajęczyny $P(O_{x_n}, O_{x_p})$ dla zbioru źródeł wiedzy $P_{\mathcal{KS}}$

Sygnaturą podrzędna s_p osobnika o dla pajęczyny $P(O_{x_n}, O_{x_p})$ nazywamy sygnaturę osobnika wyznaczoną dla terminologii podrzędnej O_{x_p} dla źródeł wiedzy $P_{\mathcal{KS}}$ i równą sygnaturze złożonej $s_c(o, f^{-1}(O_{x_p})[\mathcal{H}_{\mathcal{T}}, P_{\mathcal{KS}}])$ osobnika o w węźle $f^{-1}(O_{x_p})$ w hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$ dla źródeł wiedzy ze zbioru $P_{\mathcal{KS}}$ (patrz definicja 6.5).

Definicja 6.4 Sygnatura prosta $s_s(o, v[\mathcal{H}_T, P_{KS}])$ osobnika o w węźle v w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS}

Sygnaturą prostą osobnika o w węźle v w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS} nazywamy sygnaturę s_s równą przecięciu sygnatur osobnika o wyznaczonych dla poszczególnych źródeł wiedzy $KS \in P_{KS}$ i przypisanych do węzła v .

Definicja 6.5 Sygnatura złożona $s_c(o, v[\mathcal{H}_T, P_{KS}])$ osobnika o w węźle v w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS}

Niech $f(v) = O_{x_v}$, a $S_P = \{P: P(O_{x_v}, O_{x_p})\}$ jest zbiorem pajęczyn, w których O_{x_v} jest terminologią nadrzędną. Dodatkowo $S_{O_{x_p}} = \{O_x: P(O_{x_v}, O_x) \in S_P\}$.
Sygnaturą $s_c(o, v[\mathcal{H}_T, P_{KS}])$ nazywamy przecięcie sygnatury prostej $s_s(o, v[\mathcal{H}_T, P_{KS}])$ z sygnaturami podrzędnymi $s_p[o, P(O_{x_n}, O_{x_p}), P_{KS}]$, gdzie $O_{x_p} \in S_{O_{x_p}}$.

Przykład 6.2 cd.

W analizowanym przykładzie $P_{KS} = \{KS_1, KS_2, KS_3, KS_4, KS_5\}$. Sygnatury dla osobnika Linguini są pokazane kolejno poniżej.

$s_s(\text{Linguini}, v_3[\mathcal{H}_T, P_{KS}]) = 001$ oraz $s_c(\text{Linguini}, v_3[\mathcal{H}_T, P_{KS}]) = 001$ (zbiór pajęczyn S_P jest pusty),

$s_s(\text{Linguini}, v_4[\mathcal{H}_T, P_{KS}]) = 001$ oraz $s_c(\text{Linguini}, v_4[\mathcal{H}_T, P_{KS}]) = 001$ (zbiór pajęczyn S_P jest pusty),

$s_s(\text{Linguini}, v_2[\mathcal{H}_T, P_{KS}]) = 0000110$ oraz

$s_c(\text{Linguini}, v_4[\mathcal{H}_T, P_{KS}]) = s_s(\text{Linguini}, v_2[\mathcal{H}_T, P_{KS}]) \wedge s_p[\text{Linguini}, P(O_{x_4}, O_{x_2}), P_{KS}] \wedge s_p[\text{Linguini}, P(O_{x_4}, O_{x_3}), P_{KS}] = 0000110 \wedge 0001010 \wedge 0000110 = 0000010$

Poniżej przedstawiono algorytm SPIDER wyznaczający sygnaturę złożoną $s_c(o, v[\mathcal{H}_T, P_{KS}])$ osobnika o w węźle v w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS}

Algorytm 6.2 Algorytm SPIDER

Wejście: Węzeł v w poprawnej hierarchii terminologii \mathcal{H}_T wzbogaconej o odwzorowania w postaci pajęczyn, osobnik o , zbiór źródeł wiedzy P_{KS}

Wyjście: Sygnatura złożona $s_c(o, v[\mathcal{H}_T, P_{KS}])$ osobnika o w węźle v w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS}

1. Usuń z kopii hierarchii terminologii \mathcal{H}_T (CH_T) zbiór S_v wszystkich węzłów, które nie należą do ścieżki od pewnego węzła v_k , dla którego istnieje $KS \in P_{KS}$, do węzła v :

$$S_v = \{v_i: \neg \exists v_k (KS = \langle \langle f(v_k), O_s \rangle, Q_O \rangle \wedge f(v_k) \in P_{KS} \wedge \exists p_{v_k v_i} \wedge \exists p_{v_i v})\},$$
gdzie $p_{v_i v_j}$ oznacza ścieżkę z węzła v_i do v_j ,
2. Z CH_T usuń wszystkie $KS = \langle \langle f(v_k), O_s \rangle, Q_O \rangle \wedge f(v_k) \notin P_{KS}$.

3. Dla każdego węzła z CH_T wylicz s_s dla osobnika o (jeśli w danym węźle istnieje więcej niż jeden KS , sygnatura jest przecięciem sygnatur dla osobnika o dla każdego źródła wiedzy przypisanego do przetwarzanego węzła, jeśli w danym węźle już nie istnieje KS , sygnatura jest równa sygnaturze konceptu uniwersalnego).
4. Oblicz rekurencyjnie dla każdego węzła v_i z CH_T sygnaturę s_c jako przecięcie sygnatury prostej w przetwarzanym węźle i sygnatury nadrzędnej dla osobnika o dla pajęczyn $P[f(v_i), O_{x_p}]$.
5. Zwróć sygnaturę s_c osobnika o dla węzła v .

6.4 Podsumowanie

Zastosowanie kartograficznej reprezentacji wiedzy do budowania odwzorowań pomiędzy terminologiami w metodzie CMC ma swoje daleko idące konsekwencje. Główna zaleta takiego podejścia to połączenie zalet metod LAV i GAV. W metodzie GAV terminy zdefiniowane w terminologii nadrzędnej są wyrażane w terminach zdefiniowanych w terminologii podrzędnej. W metodzie LAV przyjęto odwrotny sposób opisywania zależności: terminy z terminologii podrzędnej są definiowane w terminach terminologii nadrzędnej. Pajęczyna zaproponowana w metodzie CMC buduje odwzorowanie dwukierunkowe. Wybierając dowolny obszar z terminologii nadrzędnej, możemy zawsze jednoznacznie wskazać obszary z terminologii podrzędnej, które spełniają dokładnie jeden z następujących warunków:

- jeśli osobnik należy do obszaru podrzędnego, to na pewno należy do obszaru nadrzędnego;
- jeśli osobnik należy do obszaru podrzędnego, to na pewno nie należy do obszaru nadrzędnego;
- jeśli osobnik należy do obszaru podrzędnego, to może należeć do obszaru nadrzędnego (nie można jednoznacznie stwierdzić, czy należy, czy też nie należy do obszaru nadrzędnego).

Analogiczna sytuacja zachodzi przy odczytywaniu pajęczyny dla obszarów z terminologii podrzędnej. Można wyróżnić następujące zalety takiego podejścia:

- istniejące pajęczyny są automatycznie aktualizowane w wyniku zmian w terminologii nadrzędnej i terminologii podrzędnej (ręczna modyfikacja jest potrzebna tylko w przypadku, kiedy zmiany dotyczą zmian semantyki terminów zdefiniowanych w terminologiach lub dodania nowych terminów lub aksjomatów);
- ręcznie dodawane są nowe pajęczyny, których semantyka wynika z dodania nowych terminów i aksjomatów do terminologii;
- łatwość przetwarzania odwzorowań metodami sygnaturowymi.

Zastosowanie metody kartograficznej wnosi również do metody pewne ograniczenia. Ograniczenia te wynikają bezpośrednio z ograniczeń samej metody kartograficznej (w jej aktualnym stopniu zaawansowania). Najważniejszym z nich jest ograniczenie ekspresywności języka definiowania terminologii nadrzędnej. Jeśli pewna terminologia O_x jest nadrzędna względem dowolnej innej terminologii, to nie może definiować konceptów postaci $\exists R.C$ takich, że R lub C są zdefiniowane w terminologiach podrzędnych i koncept $\exists R.C$ nie jest zdefiniowany w żadnej terminologii podrzędnej. Dla konceptów takiej postaci nie jest możliwe zbudowanie poprawnej pajęczyny. Wynika to z niedostatku metody kartograficznej w obsłudze zapytań postaci $\exists R.C$ typu *ad hoc* [WALO2008].

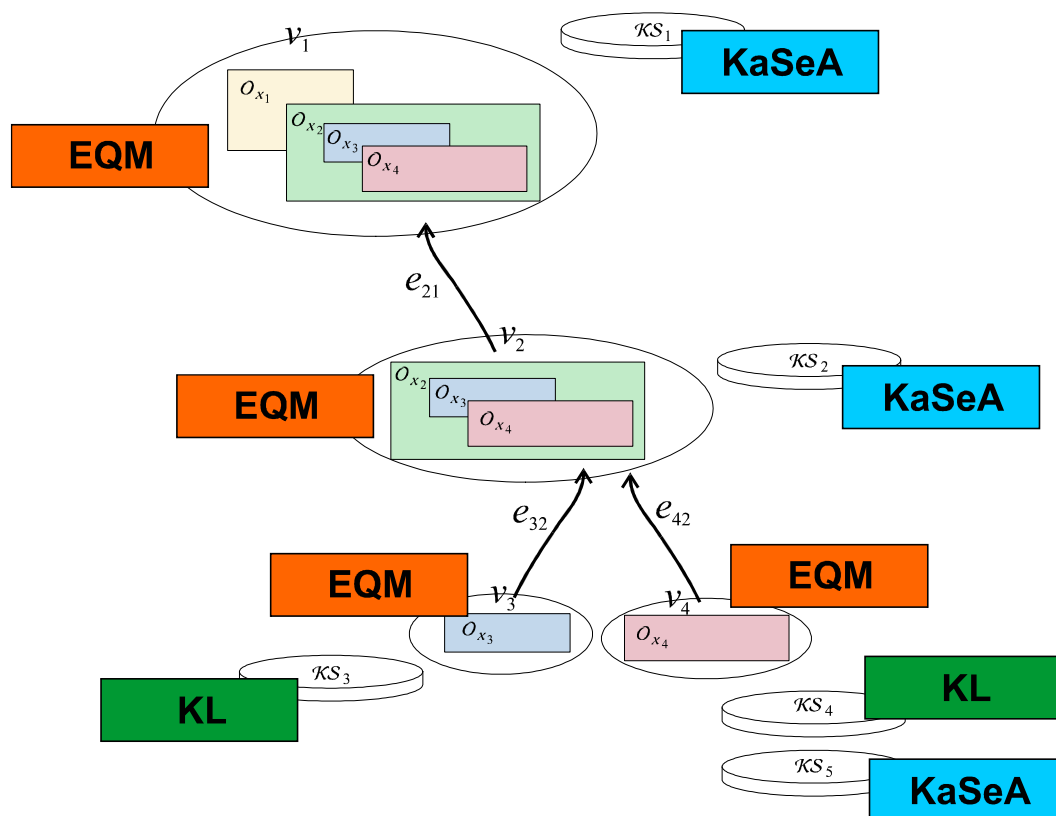
Metody wnioskowania z wykorzystaniem algorytmu SPIDER podlegają tym samym ograniczeniom co metody wnioskowania z danych opisanych kartograficznie. Odnosi się to również do pełności wnioskowania. Wnioskowanie jest zawsze poprawne, jakkolwiek może nie być pełne.

W rozdziale szóstym wraz z rozdziałem poprzednim opisano metody i algorytmy służące wytworzeniu zamodelowanego systemu integracji wiedzy. Tym samym cel drugi rozprawy został osiągnięty.

Rozdział 7. Zastosowanie metod integracji wiedzy w systemie KISS – Knowledge Integration Support System

7.1 Ogólna architektura systemu KISS (*Knowledge Integration Support System*)

W tym rozdziale zostanie wprowadzony system KISS, który został zbudowany zgodnie z modelem systemu integracji wiedzy i metodami integracji wiedzy przedstawionymi w niniejszej rozprawie. W tym rozdziale zostaną opisane poszczególne komponenty systemu, ich odpowiedzialność za zadania w ramach systemu KISS oraz wykorzystane metody integracji wiedzy zaproponowane w niniejszej rozprawie i zaimplementowane w komponentach systemu KISS. Rysunek 7.1 przedstawia przykładową hierarchię terminologii i źródeł wiedzy oraz architekturę systemu KISS w odniesieniu do tej hierarchii.



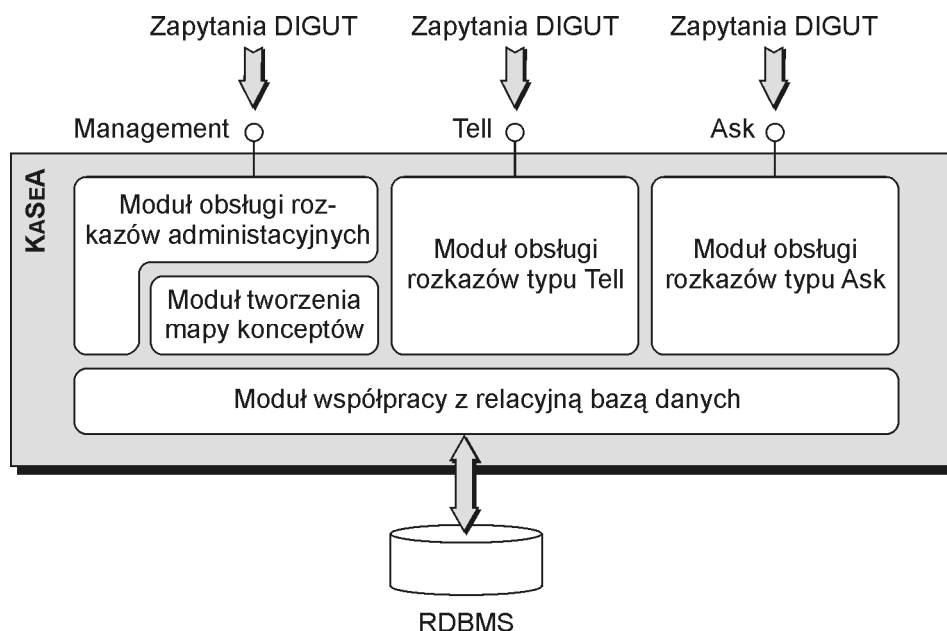
Rysunek 7.1. Komponenty systemu KISS w odniesieniu do hierarchii terminologii i źródeł wiedzy

Na rysunku 7.1 zostały wyróżnione trzy podstawowe komponenty systemu KISS:

- KaSeA (*Knowledge Signature Analyser*) – źródło wiedzy odpowiedzialne za zarządzanie wiedzą ontologiczną zapisaną w postaci zbioru asercji;
- KL (*Knowledge Layer*) – źródło wiedzy odpowiedzialne za zarządzanie wiedzą zapisaną w zewnętrznych źródłach danych;
- EQM (*External Query Manager*) – źródło wiedzy odpowiedzialne za przetwarzanie zapytań do źródeł wiedzy.

7.2 KaSeA (*Knowledge Signature Analyser*)

System KaSeA jest bazą wiedzy ontologicznej zaimplementowaną w ramach projektu 6FP EU PIPS (ang. *Personalised Information Platform for Life and Health Services*) [PIPS2007] i rozprawy doktorskiej Wojciecha Waloszka [WALO2008]. Wprowadzenie do systemu KaSeA oraz ogólnych założeń dotyczących systemu zostało przedstawione w rozdziale 4. Ogólna architektura systemu KaSeA została przedstawiona na rysunku 7.2.



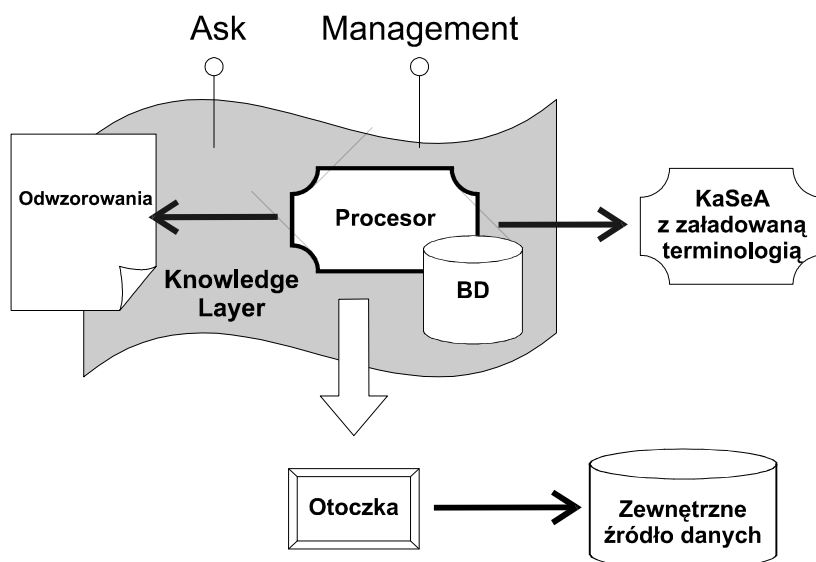
Rysunek 7.2. Architektura podsystemu KaSeA [WALO2008]

KaSeA udostępnia trzy interfejsy: *Management*, *Tell* i *Ask*. Interfejs *Management* służy do obsługi rozkazów administracyjnych (np.: tworzenie i usuwanie nowej bazy wiedzy). Za wykonywanie rozkazów z interfejsu *Management* odpowiedzialny jest moduł obsługi rozkazów administracyjnych. Drugi z obsługiwanych przez system KaSeA interfejsów to interfejs *Tell* służący do wczytywania do bazy wiedzy aksjomatów terminologicznych i zbioru asercji o osobnikach. Podczas wczytywania zbioru aksjomatów terminologicznych jest tworzona mapa konceptów, za co odpowiedzialny jest moduł tworzenia mapy konceptów. Ostatni z interfejsów to interfejs *Ask* służący do zadawania zapytań do bazy wiedzy. Za wykonywanie rozkazów z interfejsu *Ask* odpowiedzialny jest moduł obsługi rozkazów typu *Ask*. Wszystkie wymienione moduły korzystają również z funkcjonalności modułu współpracy z relacyjną bazą danych, jako że system KaSeA zarządza bazą wiedzy, która w całości jest przechowywana w relacyjnej bazie danych. Aktualna implementacja systemu KaSeA wykorzystuje system Oracle 10g. Wszystkie interfejsy systemu KaSeA są to jednolite

interfejsy zaimplementowane w języku DIGUT (*Description Logic Interface by Gdańsk University of Technology*) [DIGUT]. Język DIGUT powstał na podstawie interfejsu DIG 1.1 [BECH2003], rozszerzając go o nowe konstrukcje oraz uściślając konstrukcje istniejące.

7.3 KL (*Knowledge Layer*)

System KL dostarcza usług wnioskowania z danych zapisanych w zewnętrznych źródłach danych. W systemie KL użytkownik jest nieświadomy, gdzie faktycznie znajdują się dane, na podstawie których system generuje odpowiedź. W systemie KL można wyróżnić dwa interfejsy: interfejs *Ask* i interfejs *Management*. Zapytania w postaci rozkazów interfejsu *Ask* zadawane są do systemu KL dokładnie w taki sam sposób, jak do systemu KaSeA, tj. poprzez interfejs DIGUT. Interfejs *Management* jest odpowiedzialny za zarządzanie odwzorowaniami, które przechowują semantyczny opis danych zapisanych w zewnętrznych źródłach danych. Interfejs ten w odróżnieniu od interfejsu *Management* systemu KaSeA nie jest interfejsem DIGUT; jest to zbiór wywołań klas API. Interfejsy systemu KL oraz podział odpowiedzialności pomiędzy moduły systemu KL został przedstawiony na rysunku 7.3.



Rysunek 7.3. Architektura podsystemu KL

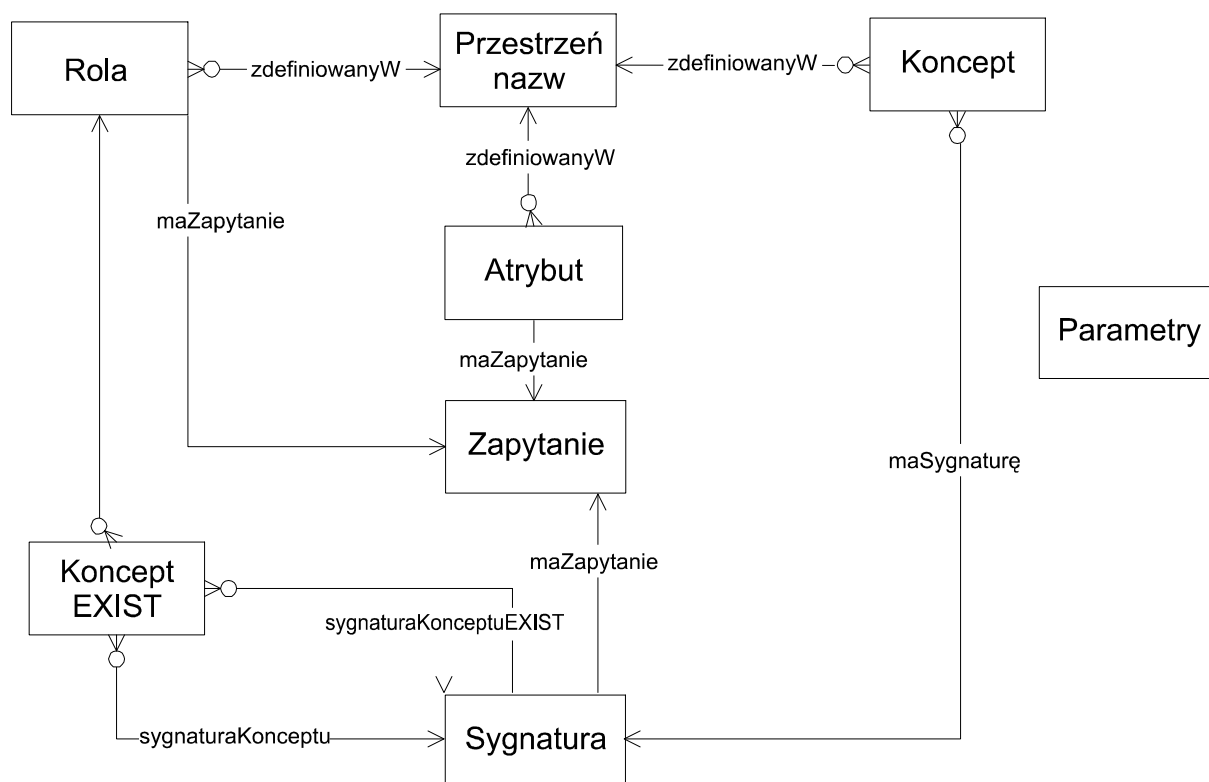
W systemie KL odwzorowania są dostarczane w postaci plików XML. Odwzorowania, o których mowa, to zbiór poprawnych i kompletnych odwzorowań conceptów, ról i atrybutów zdefiniowanych w metodzie SED tworzenia semantycznego opisu danych. Odwzorowania ról i atrybutów oraz odwzorowania conceptów przetransformowanych na odwzorowania sygnatur są zapisywane w bazie danych (BD) przez moduł Procesora. System KL współpracuje z systemem KaSeA, do którego załadowana została terminologia, w której terminach zostały odwzorowane zewnętrzne źródła danych. Podczas ładowania terminologii wyznaczone są wszystkie sygnatury dla conceptów używane później w procesie obsługi zapytań skierowanych do systemu KL. W odwzorowaniach, zgodnie z założeniami metody SED, niejawnie przechowywane są asercje o osobnikach. Aksjomaty terminologiczne ładowane są do systemu KaSeA, z którego korzysta system KL. Dlatego nie ma potrzeby wyspecyfikowania interfejsu *Tell* dla systemu KL.

Kluczowym modulem systemu KL jest Procesor, który jest odpowiedzialny za:

- transformację odwzorowań zdefiniowanych w pliku na odwzorowania w terminach sygnatur,

- zachowanie tych odwzorowań w bazie danych modułu,
- odpowiadanie na pytania zadane w terminach opisanych wspomnianą wcześniej terminologią.

W module Procesora zaimplementowane są więc również wszystkie algorytmy wnioskowania zdefiniowane w metodzie RED. Ostatnim z modułów systemu KL jest moduł relacyjnej bazy danych. Schemat relacyjnej bazy danych jest podobny do schematu zdefiniowanego dla systemu KaSeA [WALO2007]. Główną różnicą jest fakt, że w bazie danych KL nie ma informacji o osobnikach: ich nazwach, rolach i atrybutach. Informacja ta jest zawarta niejawnie w tabeli przechowującej informacje o zapytaniach. Diagram ERD dla bazy danych systemu KL jest pokazany na rysunku 7.4



Rysunek 7.4. Diagram ERD bazy danych systemu KL

Najważniejsze encje zaznaczone na schemacie ERD to: *Koncept*, *Atrybut*, *Rola*, *Sygnatura* i *Zapytanie*. Encje *Koncept*, *Atrybut* i *Rola* przechowują odpowiednio informacje o odwzorowanych conceptach, atrybutach i rolach. Encje *Atrybut* i *Rola* są powiązane bezpośrednio z encją *Zapytanie*, w taki sposób że: dla każdego atrybutu istnieje związek z zapytaniem zwracającym listę par: osobnik, wartość atrybutu, a dla każdej roli istnieje związek z zapytaniem zwracającym pary osobników. Koncepty nie są bezpośrednio powiązane z zapytaniami, ale przez encję *Sygnatury*. W bazie danych znajduje się jeszcze jedna encja *KonceptEXIST*, w której znajdują się koncepty postaci $\exists R.C$. Encja *KonceptEXIST* jest powiązana z encją *Sygnatura* dwoma związkami. Związek *sygnaturaKonceptu* wskazuje na sygnaturę konceptu C , zaś związek *sygnaturaKonceptuEXIST* wskazuje na sygnaturę konceptu $\exists R.C$. Opis poszczególnych encji został zawarty w tabeli 7.1.

W encji *Sygnatura* znajdują się dodatkowe atrybuty *hash* i *sek_i*, które zostały wprowadzone w celu zwiększenia efektywności wyszukiwania potrzebnych atrybutów. Zastosowanie tych atrybutów zostało szczegółowo opisane w [WALO2005]. Encja *Parametry*

przechowuje między innymi długość sygnatury (która jest stała dla jednej terminologii), a encja *PrzestrzeńNazw* przechowuje przestrzenie nazw dla terminów zdefiniowanych w ontologii.

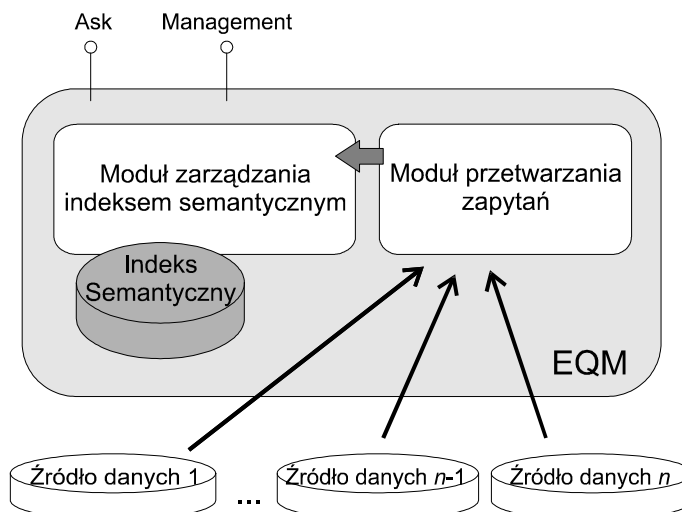
Encje	Zawartość	Atrybuty encji
<i>Koncept</i>	Informacje o odwzorowanych konceptach	<i>id</i> – identyfikator konceptu <i>nazwa</i> – lokalna nazwa konceptu
<i>Rola</i>	Informacje o odwzorowanych rolach	<i>id</i> – identyfikator roli <i>nazwa</i> – lokalna nazwa roli
<i>Atrybut</i>	Informacje o odwzorowanych atrybutach	<i>id</i> – identyfikator atrybutu <i>nazwa</i> – lokalna nazwa atrybutu
<i>Zapytanie</i>	Zapytania zrozumiałe przez otoczkę danego typu	<i>id</i> – identyfikator zapytania <i>tresc</i> – treść zapytania
<i>Sygnatura</i>	Sygnatury zawarte w odwzorowaniach sygnatur	<i>id</i> – identyfikator sygnatury, <i>hash</i> – kod hash sygnatury, <i>sek_i</i> – liczba jedynek w kolejnych sekcjach sygnatury
<i>KonceptEXIST</i>	Koncepty typu $\exists R.C$	Brak atrybutów
<i>PrzestrzeńNazw</i>	Przestrzenie nazw	<i>id</i> – identyfikator przestrzeni nazw <i>uri</i> – URI przestrzeni nazw
<i>Parametry</i>	Parametry terminologii załadowanej do systemu KaSeA	<i>nazwa</i> – nazwa parametru <i>wartość</i> – wartość parametru

Tabela 7.1. Opis encji bazy danych systemu KL

System KL komunikuje się z dwoma zewnętrznymi systemami: systemem KaSeA oraz otoczką. Otoczką potrafi wykonywać zapytania kierowane do zewnętrznego źródła danych. Zewnętrzne źródło danych może być dowolnym źródłem strukturalnym lub półstrukturalnym, co wynika z niezależności metody SED od typu źródeł danych (SQL, XML, CSV, XLS, MDB i in.). Jedynym wymaganiem jest zdefiniowanie odpowiednich zapytań i umieszczenie ich w pliku z odwzorowaniami. Taka elastyczność wymaga jednak spełnienia jednego warunku: musi istnieć język zapytań dla dowolnego typu źródła danych, dla którego są budowane odwzorowania. Język ten musi spełniać pewne wymagania (na przykład musi istnieć możliwość zdefiniowania zapytania złożonego jako sumy kilku zapytań prostych). To za pomocą tego języka muszą zostać wyrażone zapytania znajdujące się w pliku z odwzorowaniami. Oczywiście jest, że nie dla wszystkich wymienionych wcześniej typów źródeł istnieją takie języki. Dlatego też Otoczką jest modulem potrafiącym takie zapytania wykonać. Przykładowo dla danych relacyjnych rolę otoczki spełnia RDBMS, który wykonuje zapytania zdefiniowane w języku SQL.

7.4 EQM (*External Query Manager*)

Kolejnym komponentem systemu KISS jest moduł *External Query Manager* (EQM) odpowiedzialny za przetwarzanie zapytań skierowanych do źródeł wiedzy. W implementacji dla potrzeb systemu PIPS istnieje jedna instancja systemu EQM dla wszystkich węzłów hierarchii terminologii. Architektura systemu EQM została przedstawiona na rysunku 7.5.



Rysunek 7.5. Architektura podsystemu EQM

System EQM udostępnia dwa interfejsy *Ask* i *Management*. Analogicznie jak w przypadku systemu KL system EQM nie udostępnia interfejsu *Tell*. Wiedza terminologiczna jest zapisana w systemach KaSeA, zaś wiedza o osobnikach w systemach KaSeA lub w zewnętrznych źródłach danych. Wiedza ta jest ładowana do tych systemów za pomocą ich własnych interfejsów. Interfejs *Ask* systemu EQM jest analogiczny, jak w przypadku interfejsów *Ask* systemu KaSeA i KL – jest to interfejs w języku DIGUT. Jedyną różnicą jest wzbogacenie interfejsu o parametr określający grupę, do której jest zadawane zapytanie. Przez grupę w systemie EQM rozumiemy parę: terminologia i zbiór źródeł wiedzy. W odniesieniu do definicji zapytania do systemu KISS grupa odpowiada parze (O_x, P_{KS}) .

Za wykonanie zapytań wyrażonych w interfejsie *Ask* odpowiedzialny jest moduł przetwarzania zapytań. Moduł przetwarzania zapytań przekształca zapytania w zbiór zapytań do źródeł wiedzy na podstawie indeksu semantycznego. W indeksie semantycznym w sposób niejawni przechowywane są pajęczyny. W modelu systemu integracji wiedzy przedstawionym w niniejszej rozprawie każde źródło wiedzy jest opisane terminologią \mathcal{T} znajdującą się w pewnym węźle hierarchii terminologii $\mathcal{H}_{\mathcal{T}}$. Za pomocą pajęczyny każdy termin z terminologii \mathcal{T} może zostać przekształcony w odpowiadający mu termin z terminologii \mathcal{T}_i zawierającej terminologię \mathcal{T} ($\mathcal{T}_i \gg \mathcal{T}$). W systemie EQM pajęczyna w postaci odwzorowań sygnatur z terminologii nadrzędnej do terminologii podrzędnej nie jest jawnie przechowywana. Optymalizacja polega na przypisaniu źródła wiedzy opisanego terminologią \mathcal{T} do węzłów hierarchii terminologii, których terminologia \mathcal{T}_i zawiera terminologię \mathcal{T} . Przy takim rozwiązaniu zaimplementowany w module Moduł przetwarzania zapytań algorytm SPIDER nie wymaga przejrzenia hierarchii źródeł wiedzy, ograniczając zadanie zapytania do źródeł wiedzy przypisanych do węzła w hierarchii, do którego zostało skierowane zapytanie (określonego przez parametr *grupa*). Interfejsem przepływu informacji pomiędzy

systemem EQM a źródłami wiedzy jest interfejs *Ask*. Szczegółowe zasady łączenia odpowiedzi w języku DIGUT z różnych źródeł wiedzy zastosowane w algorytmie SPIDER w systemie EQM zostały opisane w [ZAWA2007a].

7.5 Podsumowanie

System KISS został zaimplementowany w ramach projektu PIPS przez grupę KMG (*Knowledge Management Group at Gdańsk University of Technology*) i firmę Atena. W niniejszym rozdziale omówiono architekturę systemu KISS, ze szczególnym naciskiem na zastosowanie przedstawionych w niniejszej rozprawie metod do wytworzenia podsystemów systemu KISS. Rozdział ten wskazuje na praktyczne wykorzystanie zaproponowanych w rozprawie metod i algorytmów i jest pierwszym krokiem do spełnienia celu trzeciego niniejszej rozprawy: wykazanie, że opracowane algorytmy i metody sprawdzają się w praktyce. Dopełnieniem niniejszego rozdziału jest kolejny rozdział 8, w którym zostało opisane zastosowanie systemu KISS w rzeczywistym systemie integracji wiedzy dotyczącej odżywiania oraz ocena przedstawionych metod na podstawie testów wydajności przeprowadzonych dla tego systemu.

Rozdział 8. Ocena przedstawionych metod integracji wiedzy

8.1 System PIPS (*Personalised Information Platform for Life and Health Services*)

System KISS został zaimplementowany w ramach prac nad projektem PIPS (*Personalised Information Platform for Life and Health Services*), objętym 6. Programem Ramowym Unii Europejskiej, obszar *Information Society Technologies*, priorytet „e-Health”. Projekt ten, zainicjowany 1.01.2004 r., był realizowany jako projekt zintegrowany (*Integrated Project*) przez 17 partnerów z Europy, Kanady i Chin. Instytucją koordynującą była fundacja San Raffaele del Monte Tabor z Mediolanu, a uczestnikami były wyższe uczelnie i instytuty naukowo-badawcze (z Polski: Politechnika Gdańska), firmy informatyczne (z Polski: firma Atena z Sopotu), producenci leków, producenci sprzętu medycznego i in. Głównym celem projektu było opracowanie infrastruktury informatycznej ukierunkowanej na indywidualnych obywateli, wspomagającej ochronę ich zdrowia i utrzymywanie zdrowego stylu życia.

System PIPS w swoich założeniach ma służyć zarówno lekarzom, pacjentom, jak i zwykłym, zdrowym obywatelom. Lekarze są wspomagani poprzez uzyskiwanie informacji pomagających w diagnostyce i terapii. Informacje te obejmują dane o pacjencie, w tym jego dane medyczne, opisy leków z informacjami o ewentualnych przeciwwskazaniach i niepożądanych interakcjach, a także wszelkie inne informacje medyczne pomocne we właściwej diagnostyce i terapii. Pacjenci zagrożeni szczególnie niebezpiecznymi chorobami są monitorowani za pomocą urządzeń telemedycznych i w razie potrzeby informowani o sytuacjach krytycznych. Pacjenci uzyskują również wskazówki co do postępowania w takich sytuacjach. Zwykli, zdrowi obywatele uzyskują informacje dotyczące utrzymania dobrego zdrowia, w tym wytyczne odnośnie diety, zdrowego stylu życia itd. Istotne jest, że wytyczne te nie mają charakteru ogólnego, ale są dostosowane do konkretnego człowieka – jego historii, upodobań, przyzwyczajeń, miejsca zamieszkania itd.

Osiągnięcie celów systemu PIPS było uzależnione od dwóch zasadniczych czynników. Pierwszy dotyczy włączenia omawianego systemu w strukturę aktualnie istniejących procesów biznesowych związanych z szeroko pojętą ochroną zdrowia i jego społecznej akceptacji. Drugi dotyczył realizacji zaawansowanych funkcji systemu.

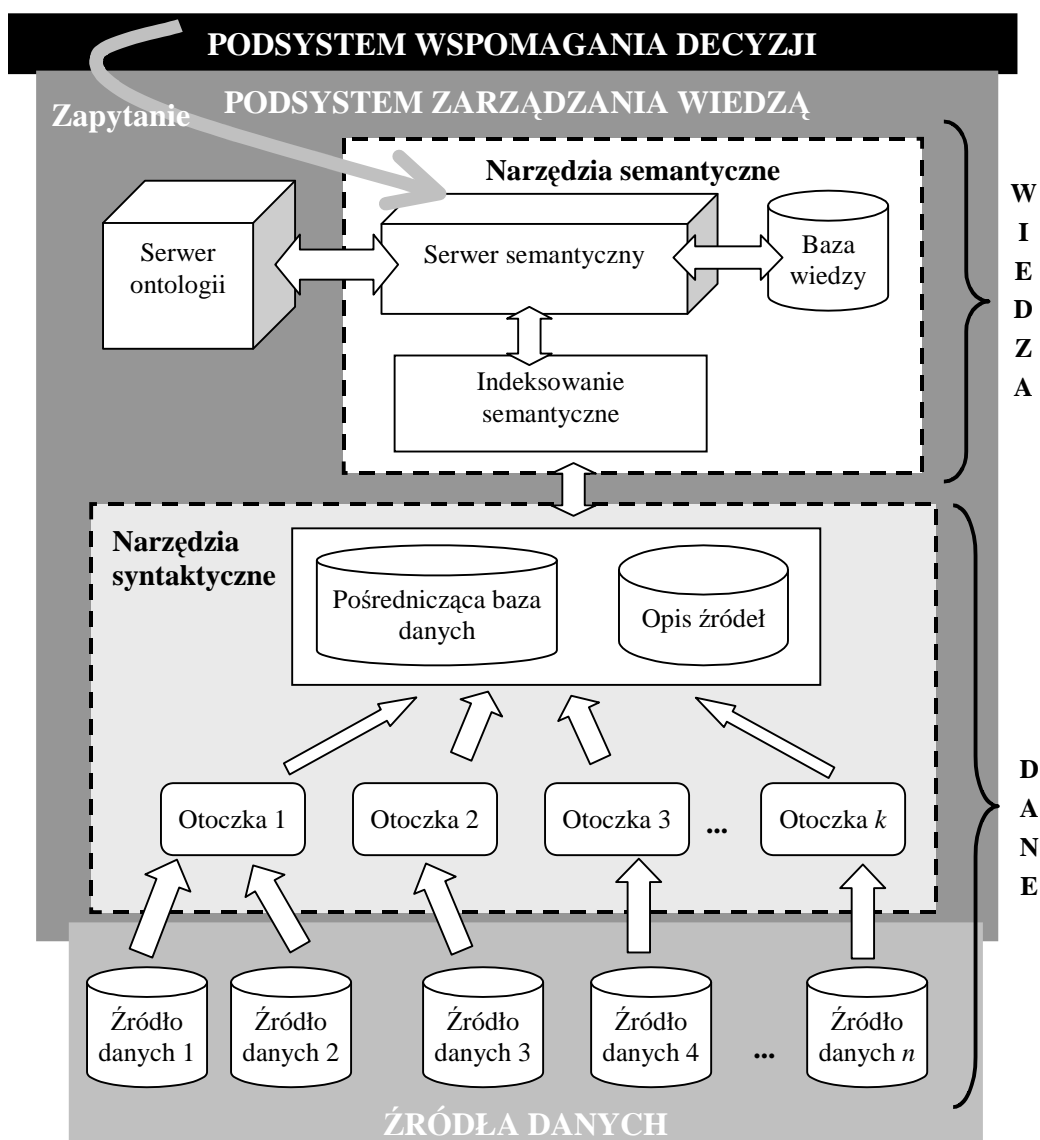
Realizacja zaawansowanych funkcji systemu PIPS jest w szczególności uzależniona od dysponowania i zarządzania odpowiednio obszerną, wielopłaszczyznową wiedzą dziedzinową. Wiedza ta obejmuje m.in. informacje o lekach i interakcjach pomiędzy lekami, informacje wspomagające diagnostykę (objawy chorób), informacje o żywności i interakcjach pomiędzy żywnością a lekami, wskazówki dietetyczne, a także informacje o charakterze

mniej krytycznym, jak np. miejsca, gdzie można nabyć zalecaną żywność lub gdzie można „bezpiecznie” pójść na obiad.

PIPS nie tylko przechowuje tę wiedzę i udostępnia ją na żądanie, ale także wywodzi z niej wnioski, aby móc odpowiadać na nietrywialne pytania kierowane przez lekarzy, pacjentów i innych jego udziałowców. Za zadania te w systemie PIPS odpowiada podsystem zarządzania wiedzą (ang. *Knowledge Management Subsystem*) [GOCZ2004], [GOCZ2005b].

Tak więc w systemie PIPS wiedza jest kluczowym zasobem, a jednym z głównych celów tego projektu było opracowanie nowatorskiego, bezpiecznego dla użytkowników systemu zarządzania wiedzą dotyczącą szeroko rozumianego zdrowia. Cel ten mógł być osiągnięty tylko poprzez wykorzystanie najnowszych osiągnięć z dziedziny inżynierii ontologii i zarządzania ontologiami, wnioskowania oraz integracji heterogenicznych źródeł danych. W te osiągnięcia w sposób szczególny wpisują się opracowanie systemu wnioskującego KaSeA [WALO2008] oraz metody integracji wiedzy przedstawione w niniejszej rozprawie.

Ogólna idea systemu zarządzania wiedzą, którego bazę stanowi system integracji wiedzy, została zaproponowana już w 2004 r. w [GOCZ2004] i zachowana w podobnym kształcie również w ostatecznej wersji systemu PIPS.



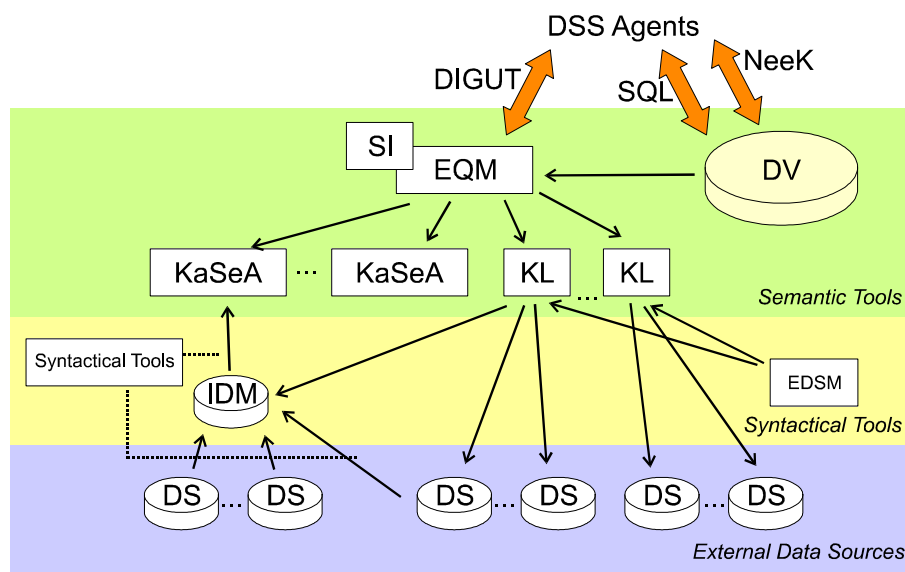
Rysunek 8.1. Pierwotna architektura systemu zarządzania wiedzą w systemie PIPS [GOCZ2004]

Rysunek 8.1 przedstawia architekturę systemu zarządzania wiedzą w systemie PIPS. Podsystem wspomagania decyzji jest bezpośrednim klientem podsystemu zarządzania wiedzą i jest odpowiedzialny za interpretację potrzeb użytkownika. Podsystem wspomagania decyzji wyraża potrzebę użytkownika w postaci zapytania skierowanego do podsystemu zarządzania wiedzą. Zwrócona z tego podsystemu odpowiedź zostaje przekształcona w spersonalizowaną odpowiedź dla konkretnego użytkownika.

Serwer ontologii, narzędzia semantyczne i narzędzia syntaktyczne stanowią podsystem zarządzania wiedzą. Serwer ontologii zarządza schematem wiedzy opisującym potrzebne fragmenty wiedzy z dziedziny zdrowia, medycyny, dietetyki itd. Dodatkowo udostępnia zaawansowane usługi wnioskowania, zarówno na potrzeby zewnętrznych klientów podsystemu, jak i na potrzeby integracji danych. Serwer semantyczny pełni rolę procesora zapytań odpowiedzialnego za przetwarzanie zapytania, jego optymalizację, planowanie wykonania i zarządzanie jego wykonaniem [GOCZ2005c], [GOCZ2005d]. Indeksowanie semantyczne zarządza opisami semantyki źródeł w terminach zrozumiałych dla serwera semantycznego. Analogicznie jak w przypadku systemów integracji danych, otoczki umożliwiają dostęp do konkretnych typów źródeł danych. Baza wiedzy przechowuje dane krytyczne z punktu widzenia zarządzania wiedzą z zakresu szeroko rozumianego zdrowia. Pośrednicząca baza danych pełni rolę bufora przechowującego dane zebrane przez otoczki.

W systemie KISS rolę serwera semantycznego i serwera ontologii pełnią podsystemy KaSeA i podsystem EQM. Podsystem KaSeA jest odpowiedzialny za dostarczenie usług wnioskowania i zarządzania schematem wiedzy – terminologią, zaś system EQM za zadania związane z integracją wiedzy. W pierwotnej architekturze systemu zarządzania wiedzą w projekcie PIPS wszystkie dane z zewnętrznych źródeł danych miały być dostarczane do serwera semantycznego przez pośredniczącą bazę danych. Otoczki miały pobierać dane z zewnętrznych źródeł danych i zapisywać je w jednolitym formacie w pośredniczącej bazie danych. Takie rozwiązanie pozostało w systemie KISS tylko dla danych niestrukuralnych, takich jak strony HTML. Pośrednicząca baza danych w systemie KISS jest traktowana analogicznie jak inne strukturalne lub półstrukturalne źródła danych jako zewnętrzne źródło danych dla systemu KL.

Początkowa architektura systemu zarządzania wiedzą została wzbogacona również o bazodanowy widok wiedzy (ang. *Data View*) [GOCZ2005e] i podsystem zarządzania zewnętrznymi źródłami danych EDSM (ang. *External Data Source Manager*) odpowiedzialny za aktualizację odwzorowań. Podsystem zarządzania wiedzą dostarcza podsystemowi wspomagania decyzji dwóch interfejsów: języka DIGUT [DIGUT], który bezpośrednio odpytuje narzędzia syntaktyczne, oraz bazy danych, w której jest zapisana wiedza w sposób relacyjny. Takie rozwiązanie zostało zaproponowane ze względu na specyfikę wykorzystania bazy wiedzy w systemie PIPS: zarówno jako systemu wnioskującego, jak i systemu integracji wiedzy (a więc również i danych). Wynikowa architektura podsystemu zarządzania wiedzą w projekcie PIPS uwzględniająca nowe komponenty, takie jak *Knowledge Layer*, *Data View* i *External Data Source Manager* została przedstawiona na rysunku 8.2.



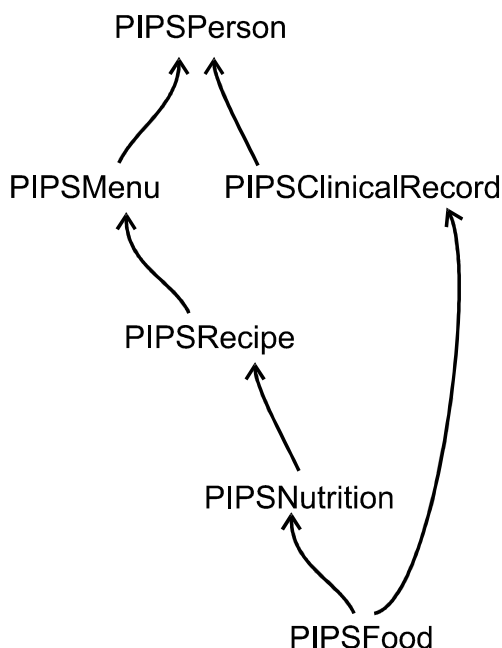
Rysunek 8.2. Wynikowa architektura systemu zarządzania wiedzą w systemie PIPS [GOCZ2004]

W systemie PIPS opracowano różne usługi wspomaganie zdrowego stylu życia i wsparcia w procesie leczenia dla lekarzy i pacjentów. Do najważniejszych z nich należą: usługa powszechnego dostępu do wiarygodnych źródeł wiedzy medycznej, usługa doboru i wspomaganie stosowania optymalnej diety, usługa różnorodnego motywowania do codziennej aktywności fizycznej oraz usługa ciągłego nadzoru zdrowia w domu [PIPS2008]. System KISS został wykorzystany w usłudze doboru i wspomaganie stosowania optymalnej diety do zarządzania przepisami i definiowania menu.

8.2 Usługa doboru i wspomaganie stosowania optymalnej diety

Omawiana usługa polega na wykorzystaniu nowoczesnych technologii informacyjnych i istniejących baz danych dotyczących produktów spożywczych do tego, by w sposób ciągły kontrolować skład i dobór diety na wszystkich etapach dotyczących żywienia: w trakcie zakupów, przy wyborze posiłków w restauracji i punktach zbiorowego żywienia, w trakcie przygotowywania posiłków w domu, i to w sposób uwzględniający stan naszego zdrowia, indywidualne ograniczenia czy zalecenia dietetyczne. Odbiorcami tej usługi są osoby podlegające ograniczeniom dietetycznym oraz osoby zainteresowane zdrowym żywieniem.

Wiedza terminologiczna w omawianej usłudze została przygotowana przez University of Liverpool jako zbiór ontologii OWL. Hierarchia terminologii została przedstawiona na rysunku 8.3.



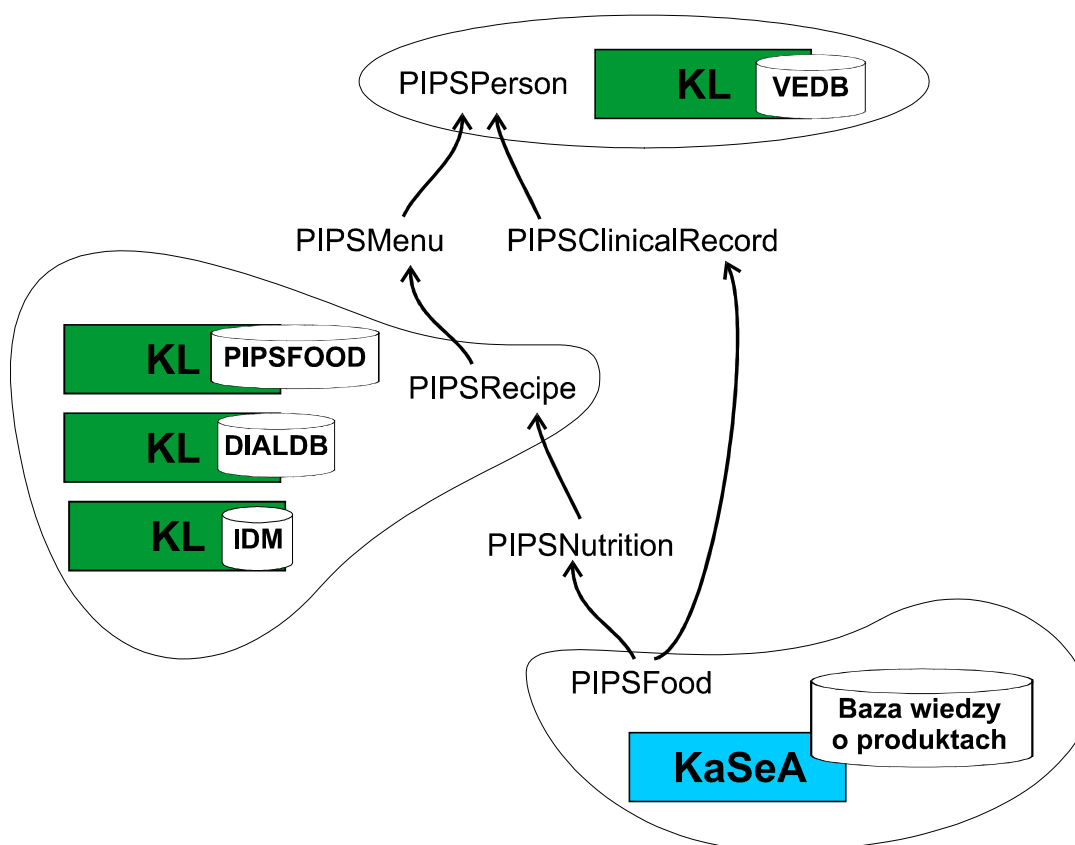
Rysunek 8.3. Hierarchia terminologii w systemie KISS dla usługi doboru i wspomaganie optymalnej diety

Opis terminologii znajduje się w tabeli 8.1.

Ontologia	Opis	Liczba konceptów	Liczba ról i atrybutów
<i>PIPSFood</i>	Schemat wiedzy przechowujący produkty i ich podział na odpowiednie grupy żywieniowe	179	3
<i>PIPSNutrition</i>	Schemat wiedzy przechowujący piramidę odżywiania	88	9
<i>PIPSRecipe</i>	Schemat wiedzy przechowujący przepisy wraz z uwzględnieniem ich wartości odżywczych	19	15
<i>PIPSMenu</i>	Schemat wiedzy opisujące wzorce menu	23	6
<i>PIPSClinicalRecord</i>	Schemat wiedzy opisujący uwarunkowania pacjenta ze szczególnym uwzględnieniem alergii i nietolerancji produktów żywieniowych	7	3
<i>PIPSPerson</i>	Schemat wiedzy przechowujący ogólne informacje o obywatelu, dane dotyczące jego upodobań kulinarnych oraz uwarunkowania zdrowotne.	25	13

Tabela 8.1. Opis ontologii w systemie KISS dla usługi doboru i wspomaganie optymalnej diety

W usłudze doboru i wspomaganie optymalnej diety integrowano wiedzę z pięciu źródeł wiedzy: jednej bazy wiedzy zarządzanej przez system KaSeA oraz czterech baz danych Oracle 10g zarządzanych przez system KL. Przypisanie odpowiednich źródeł wiedzy do odpowiednich węzłów w hierarchii terminologii zostało przedstawione na rysunku 8.4.



Rysunek 8.4. Hierarchia źródeł wiedzy w systemie KISS dla usługi doboru i wspomagania optymalnej diety

Opis poszczególnych źródeł wiedzy znajduje się w tabeli 8.2.

Źródło wiedzy	Opis	Liczba
Baza wiedzy o produktach – KaSeA	Źródło wiedzy o produktach	≈ 60 000 asercji
IDM – KL	Źródło wiedzy dla pośredniczącej bazy wiedzy	≈ 100 000 krotek
DIALDB – KL	Źródło wiedzy przepisów	≈ 7 500 krotek
PIPSFOOD – KL	Źródło wiedzy o przepisach i produktach	≈ 1 500 krotek
VEDB – KL	Źródło wiedzy o osobach	≈ 1 000 krotek

Tabela 8.2. Opis źródeł wiedzy w systemie KISS dla usługi doboru i wspomagania optymalnej diety

Dla usługi doboru i wspomagania optymalnej diety zdefiniowano listę istotnych zapytań (tzw. zapytań kompetencyjnych). Zapytania kompetencyjne zostały zdefiniowane dla trzech terminologii: PIPSFood, PIPSRecipe i PIPSPerson.

Dla terminologii PIPSFood są to następujące zapytania:

1. Podaj grupę żywieniową produktu żywieniowego.
2. Podaj warstwę piramidy żywieniowej odpowiadającej danemu produktowi żywieniowemu.
3. Podaj rekomendowany sposób podania produktu żywieniowego.

Dla terminologii PIPSRecipe są to następujące zapytania:

1. Podaj nazwę i opis danego przepisu.

2. Podaj na jaki posiłek dany przepis jest odpowiedni.
3. Podaj wartości odżywcze przepisu.
4. Podaj danie na które może składać się przepis.
5. Podaj składniki i ich ilości zastosowane w danym przepisie.
6. Podaj grupy żywieniowe, do których należą poszczególne składniki danego przepisu.
7. Podaj podstawową grupę żywieniową dla danego przepisu.

Dla terminologii PIPSPerson są to następujące zapytania:

1. Podaj zapotrzebowanie energetyczne danej osoby.
2. Podaj wzrost, wagę, płeć oraz wiek danej osoby.
3. Podaj specjalne uwarunkowania danej osoby (ciąża, kobieta karmiąca).
4. Podaj ulubione produkty danej osoby.
5. Podaj ulubione przepisy danej osoby.
6. Podaj produkty żywieniowe, których dana osoba nie lubi.
7. Podaj przepisy, których dana osoba nie lubi.
8. Podaj zbiór osobistych przepisów danej osoby.
9. Podaj wzorzec menu dla danej osoby.

W tabeli 8.3 opisano poszczególne zapytania kompetencyjne, sposób ich wykonania i uzyskany czas odpowiedzi od systemu KISS. Testy zostały wykonane na komputerze Intel Xeon 5120, 2 GB RAM.

PIPSFood	
1. Podaj grupę żywieniową produktu żywieniowego.	
<p>Zwróć typy zadanego produktu żywieniowego, które są potomkami konceptu <i>FoodGroup</i>. Zapytanie jest wykonywane w dwóch krokach:</p> <p>1.1 Znajdź typy danego produktu żywieniowego.</p> <p>1.2 Znajdź koncepty będące potomkami konceptu <i>FoodGroup</i>.</p> <p>Odpowiedź jest przecięciem zbiorów nazw konceptów zwróconych przez zapytania 1.1. i 1.2. W trakcie wykonania zapytania do systemu KISS zadawane są dwa zapytania: <i>types(produktZywieniowy)(PIPSFood)</i> i zapytanie <i>descendants(FoodGroup)(PIPSFood)</i>. Zapytanie <i>descendants</i> jest zapytaniem terminologicznym i jest przekazywane do systemu KaSeA z załadowaną terminologią PIPSPFood.</p> <p>Uśredniony czas wykonania zapytania: 0,9 s.</p>	
2. Podaj warstwę piramidy żywieniowej odpowiadającej danemu produktowi żywieniowemu.	
<p>Zwraca osobniki będące dopełnieniem roli <i>isContainedInto</i> typu <i>PyramidLayer</i> dla danego produktu żywieniowego. Zapytanie jest wykonywane w dwóch krokach:</p> <p>2.1 Znajdź osobniki będące dopełnieniem roli <i>isContainedInto</i> dla danego produktu żywieniowego.</p> <p>2.2 Zwróć wystąpienia konceptu <i>PyramidLayer</i>.</p> <p>Odpowiedź jest przecięciem nazw osobników zwróconych przez zapytania 2.1 i 2.2. W trakcie wykonania zapytania do systemu KISS zadawane są dwa zapytania o osobniki: <i>roleFillers(isContainedInto, produktZywieniowy)(PIPSFood)</i> i zapytanie <i>instances(PyramidLayer)(PIPSFood)</i>.</p> <p>Uśredniony czas wykonania zapytania: 1,5 s.</p>	

3. Podaj rekomendowany sposób podania produktu żywieniowego.

Zwraca osobniki będące dopełnieniem roli *hasServingRecommendation* dla danego produktu żywieniowego. Do systemu KISS jest zadawane jedno zapytanie *roleFillers(hasServingRecommendation, produktŻywieniowy)(PIPSFood)*.

Uśredniony czas wykonania zapytania: 1,3 s.

PIPSRecipe

1. Podaj nazwę i opis przepisu.

Zwraca wartości atrybutów *hasName* i *hasDescription* dla danego przepisu. Do systemu KISS jest zadawane jedno zapytanie *toldValues({hasName, hasDescription}, przepis)(PIPSRecipe)*.

Uśredniony czas wykonania zapytania: 0,5 s.

2. Podaj posiłek, na jaki odpowiedni jest dany przepis. .

Zwraca osobniki będące dopełnieniem roli *hasMealCategory* dla danego przepisu. Do systemu KISS zadawane jest jedno zapytanie *roleFillers(hasMealCategory, przepis)(PIPSRecipe)*.

Uśredniony czas wykonania zapytania: 0,3 s.

3. Podaj wartości odżywcze danego przepisu.

Zapytanie jest wykonywane w dwóch krokach:

3.1 Zwróć listę osobników typu *NutritionalComponentEntries* dla danego przepisu:

- Zwróć osobniki będące dopełnieniem łańcucha dwóch ról: *hasNutritionalProfile* i *contains* dla zadanego przepisu.
- Dla każdego zwróconego osobnika sprawdź, czy zwrócony osobnik jest wystąpieniem konceptu *NutritionalComponentEntry*.

3.2 Dla każdego osobnika zwróconego w zapytaniu 3.1 spytaj o dopełnienia roli *contains* i wartości atrybutu *quantity*.

Do systemu KISS zadawany jest zbiór zapytań (zależny od odpowiedzi na poprzednie zapytania): zapytanie *roleFillers(hasNutritionalProfile°contains, przepis)(PIPSRecipe)*, zbiór zapytań typu *instanceof(NutritionalComponentEntry, i_i)(PIPSRecipe)*, gdzie i_i są osobnikami należącymi do zbioru odpowiedzi na zapytanie 3.1.a, zbiór zapytań typu *roleFillers(contains, j_i)(PIPSRecipe)* i *toldValues({quantity}, j_i)(PIPSRecipe)*, gdzie j_i są osobnikami należącymi do zbioru odpowiedzi na zapytanie 3.1.

Uśredniony czas wykonania zapytania: 4,0 s.

4. Podaj danie, na które może składać się dany przepis.

Zwraca osobniki będące dopełnieniem roli *hasDishCategory* dla danego przepisu. Do systemu KISS zadawane jest jedno zapytanie *roleFillers(hasDishCategory, przepis)(PIPSRecipe)*.

Uśredniony czas wykonania zapytania: 0,3 s.

5. Podaj składniki i ich ilości zastosowane w danym przepisie.

Zapytanie jest zadawane w dwóch krokach:

5.1 Zwróć osobniki typu *FoodEntry* będące dopełnieniem roli *contains* dla danego przepisu (przecięcie zbiorów osobników zwróconych w zapytaniach 5.1.a i 5.1.b).

- a. Zwróć osobniki będące dopełnieniem roli *contains* dla danego przepisu.
- b. Zwróć wystąpienia konceptu *FoodEntry*.

5.2 Dla każdego osobnika zwróconego w zapytaniu 5.1 zwróć osobniki będące dopełnieniem roli *hasFood* i wartości atrybutów *quantity* i *hasUnitOfMeasure*.

Do systemu KISS zadawany jest zbiór zapytań (zależny od odpowiedzi na poprzednie zapytania): zapytanie *roleFillers(contains, przepis)(PIPSRecipe)*, zapytanie *instances(FoodEntry)(PIPSRecipe)* i zbiór zapytań typu *roleFillers(hasFood, i_i)(PIPSRecipe)*, *toldValues({quantity, hasUnitOfMeasure}, i_i)(PIPSRecipe)*, gdzie *i_i* są osobnikami należącymi do zbioru odpowiedzi na zapytanie 5.1.

Uśredniony czas wykonania zapytania: 3,2 s.

6. Podaj grupy żywieniowe, do których należą poszczególne składniki danego przepisu.

Zwraca osobniki będące dopełnieniem łańcucha ról: *hasServingDistributionEntry*, *hasFoodGroup* dla danego przepisu. Do systemu KISS zadawane jest jedno zapytanie: *roleFillers(hasDishCategory hasFoodGroup, przepis)(PIPSRecipe)*.

Uśredniony czas wykonania zapytania: 0,2 s.

7. Podaj podstawową grupę żywieniową dla danego przepisu.

Zwraca osobniki będące dopełnieniem roli *hasMainFoodGroup* dla danego przepisu. Do systemu KISS zadawane jest jedno zapytanie: *roleFillers(hasMainFoodGroup, przepis)(PIPSRecipe)*.

Uśredniony czas wykonania zapytania: 0,2 s.

PIPSPerson

1. Podaj zapotrzebowanie energetyczne danej osoby.

Zwraca wartości atrybutu *hasDailyEnergyExpenditure* dla danej osoby. Do systemu KISS zadawane jest jedno zapytanie: *toldValues(hasDailyEnergyExpenditure, osoba)(PIPSPerson)*.

Uśredniony czas wykonania zapytania: 0,2 s.

2. Podaj wzrost, wagę, płeć i wiek danej osoby

Zwraca wartości atrybutów *hasGender*, *hasAge*, *hasWeight* i *hasHeight* dla danej osoby. Do systemu KISS zadawane jest jedno zapytanie:

toldValues({hasGender, hasAge, hasWeight, hasHeight}, osoba)(PIPSPerson).

Uśredniony czas wykonania zapytania: 0,1 s.

3. Podaj specjalne uwarunkowania danej osoby (ciąża, kobieta karmiąca)

Zwraca osobniki będące dopełnieniem roli *hasSpecialCondition* dla danej osoby. Do systemu KISS zadawane jest jedno zapytanie *roleFillers(hasSpecialCondition, osoba)(PIPSPerson)*

Uśredniony czas wykonania zapytania: 0,2 s.

4. Podaj ulubione produkty danej osoby

Zwraca osobniki będące dopełnieniem roli *likes* typu \square_{ord} dla danej osoby. Zapytanie jest wykonywane w dwóch krokach:

4.1 Znajdź osobniki będące dopełnieniem roli *likes* dla danej osoby.

4.2 Dla każdego osobnika zwróconego w zapytaniu 4.1 sprawdź czy jest wystąpieniem konceptu \square_{ord}

Do systemu KISS zadawany jest zbiór zapytań: zapytanie *roleFillers(likes, osoba)(PIPSPerson)*, zapytania typu *instanceof(\square_{ord} , i_i)(PIPSPerson)*, gdzie i_i są osobnikami należącymi do zbioru odpowiedzi na zapytanie 4.1.

Uśredniony czas wykonania zapytania: 0,2 s.

5. Podaj ulubione przepisy danej osoby

Zwraca osobniki będące dopełnieniem roli *likes* typu *Recipe* dla danej osoby. Zapytanie jest wykonywane w dwóch krokach:

5.1 Znajdź osobniki będące dopełnieniem roli *likes* dla danej osoby.

5.2 Dla każdego osobnika zwróconego w zapytaniu 5.1 sprawdź, czy jest wystąpieniem konceptu *Recipe*.

Do systemu KISS zadawany jest zbiór zapytań: zapytanie *roleFillers(likes, osoba)(PIPSPerson)* i zapytania typu *instanceof(Recipe, i_i)(PIPSPerson)*, gdzie i_i są osobnikami należącymi do zbioru odpowiedzi na zapytanie 5.1.

Uśredniony czas wykonania zapytania: 0,9 s.

6. Podaj produkty żywieniowe, których dana osoba nie lubi

Zwraca osobniki będące dopełnieniem roli *dislikes* typu \square_{ord} dla danej osoby. Zapytanie jest wykonywane w dwóch krokach:

6.1 Znajdź osobniki będące dopełnieniem roli *dislikes* dla zadanej osoby.

6.2 Dla każdego osobnika zwróconego w zapytaniu 6.1 sprawdź czy jest wystąpieniem konceptu \square_{ord}

Do systemu KISS zadawany jest zbiór zapytań: zapytanie *roleFillers(dislikes, osoba)(PIPSPerson)* i zapytania typu *instanceof(\square_{ord} , i_i)(PIPSPerson)*, gdzie i_i są osobnikami należącymi do zbioru odpowiedzi na zapytanie 6.1.

Uśredniony czas wykonania zapytania: 0,2 s.

7. Podaj przepisy, których dana osoba nie lubi.

Zwraca osobniki będące dopełnieniem roli *dislikes* typu *Recipe* dla danej osoby. Zapytanie jest wykonywane w dwóch krokach:

7.1 Znajdź osobniki będące dopełnieniem roli *dislikes* dla zadanej osoby.

7.2 Dla każdego osobnika zwróconego w zapytaniu 7.1 sprawdź, czy jest wystąpieniem konceptu *Recipe*.

Do systemu KISS zadawany jest zbiór zapytań: zapytanie *roleFillers(dislikes, osoba)(PIPSPerson)* i zapytania typu *instanceof(Recipe, i_i)(PIPSPerson)*, gdzie i_i są osobnikami należącymi do zbioru odpowiedzi na zapytanie 7.1.

Uśredniony czas wykonania zapytania: 1,0 s.

8. Podaj zbiór osobistych przepisów danej osoby.

Zwraca osobniki będące dopełnieniem roli *hasRecipeBook* dla danej osoby. Do systemu KISS zadawane jest jedno zapytanie: *roleFillers(hasRecipeBook, osoba)(PIPSPerson)*.

Uśredniony czas wykonania zapytania: 0,2 s.

9. Podaj wzorzec menu dla danej osoby.

Zwraca osobniki będące dopełnieniem roli *hasMenuTemplate* dla danej osoby. Do systemu KISS zadawane jest jedno zapytanie: *roleFillers(hasMenuTemplate, osoba)(PIPSPerson)*.

Uśredniony czas wykonania zapytania: 0,2 s.

Tabela 8.3. Zapytania kompetencyjne w systemie KISS dla usługi doboru i wspomagania optymalnej diety

W tabeli 8.3 zostały pokazane wyniki testów systemu KISS dla usługi doboru i wspomagania optymalnej diety. Kolejne testy wydajnościowe ilustrują porównanie wydajności pomiędzy dwoma przedstawionymi w pracy źródłami wiedzy: systemem KaSeA i systemem KL, w którym zaimplementowano metody SED i RED.

Zapytanie	KaSeA	KL
wystąpienie konceptu <i>Recipe</i>	2,9 s	1,2 s
wystąpienie konceptu <i>Recipe</i> \sqcup <i>NutritionalComponentEntry</i>	3,0 s	1,8 s
wystąpienie konceptu <i>VegetablesFoodGroup</i>	2,8 s	1,6 s
sprawdzenie przynależności osobnika do konceptu <i>Recipe</i>	1,1 s	1,4 s
sprawdzenie przynależności osobnika do konceptu <i>Recipe</i> \sqcup <i>NutritionalComponentEntry</i>	1,0 s	1,8 s
wystąpienie roli <i>hasFood</i>	1,1 s	1,3 s
wystąpienie roli <i>hasDishCategory</i>	0,9 s	1,1 s
typ osobnika	1,0 s	2,6 s

Tabela 8.4. Porównanie czasów odpowiedzi źródeł wiedzy KL i KaSeA

W tabeli 8.4 zebrane zostały porównania czasów odpowiedzi na dla systemów KaSeA i KL. Eksperymenty przeprowadzone zostały na ontologii przepisów PIPSRecipe oraz na źródle danych DialDB. Każdy eksperyment został powtórzony 100 razy, a czasy zostały uśrednione. Eksperymenty przeprowadzone zostały na komputerze Intel Xeon 512, 2 GB RAM. Eksperymenty te pokazują, że czasy dla systemu KL są porównywalne lub niższe od systemu KaSeA. Jedynym wyjątkiem jest czas odpowiedzi na zapytanie o typ osobnika. Wykonanie zapytania o typ osobnika trwa zdecydowanie dłużej aniżeli dla systemu KaSeA. Wynika to stąd, że w przypadku systemu KaSeA pełna informacja o przynależności osobników do konceptów jest przechowywana w sygnaturze, a w przypadku systemu KL informacja ta jest pobierana z zewnętrznego źródła danych na drodze wykonania wszystkich zapytań zdefiniowanych w zbiorze odwzorowań konceptów.

W następnym podrozdziale przedstawiono oszacowanie złożoności obliczeniowej algorytmów MCA i SPIDER.

8.3 Analiza złożoności obliczeniowej algorytmów

Metody integracji wiedzy zaproponowane w niniejszej rozprawie bazują na dwóch podstawowych algorytmach: algorytmie MCA i algorytmie SPIDER.

Algorytm MCA bazuje na algorytmie Apriori o złożoności obliczeniowej eksponencjalnej. Złożoność algorytmu MCA w najgorszym przypadku jest równa $n!$, gdzie n jest liczbą odwzorowanych konceptów. Pomimo tego, że z punktu widzenia teoretycznego złożoność ta jest nie do przyjęcia, to testy praktyczne wskazują na praktyczną użyteczność algorytmu MCA, analogicznie jak w przypadku algorytmu Apriori. Dodatkowo, praktyczna użyteczność algorytmu MCA wynika z charakterystyki metody SED: liczba odwzorowywanych konceptów jest zazwyczaj niewielka. W analizowanym w poprzednim podrozdziale przypadku użycia systemu KISS liczba odwzorowywanych konceptów dla poszczególnych źródeł nie przekraczała 20, co jest pokazane w tabeli 8.5.

Źródło wiedzy - KL	Liczba odwzorowanych		
	konceptów	ról	atrybutów
PIPSFOOD	8	5	6
DIALDB	8	6	5
IDM	9	7	7
VEDB	19	14	12

Tabela 8.5. Liczba odwzorowanych konceptów, ról i atrybutów w systemie KISS dla usługi doboru i wspomagania stosowania optymalnej diety

Istotniejsze jednak nawet od liczby bezwzględnej odwzorowanych konceptów jest stosunek liczby konceptów odwzorowanych do liczby konceptów zdefiniowanych w terminologii. I tak w przypadku źródeł wiedzy dla zewnętrznych źródeł danych PIPSFOOD, DIALDB i IDM stosunek ten jest nie większy niż $9/286 \approx 0,03$ (286 jest liczbą konceptów zdefiniowanych w terminologii PIPSRecipe i wszystkich terminologiach do niej podrzędnych). W przypadku źródła wiedzy dla zewnętrznego źródła danych stosunek ten wynosi $19/311 \approx 0,06$. Oznacza to, że statystycznie mniej niż jeden koncept na każde 100 konceptów jest odwzorowywany.

Złożoność obliczeniowa algorytmu MCA nie przeszkadza jego praktycznemu wykorzystaniu również dlatego, że algorytm MCA uruchamiany do obliczania zapytania odpowiedniego dla konceptu (również złożonego) jest wykonywany dokładnie raz, ponieważ takie zapytanie może być zapamiętane jak odwzorowanie sygnatury niepodane jawnie.

Złożoność algorytmu SPIDER można oszacować jako $O(nm^k)$, gdzie n oznacza liczbę osobników, m oznacza maksymalną liczbę terminologii podrzędnych w jednym węźle hierarchii terminologii, zaś k oznacza długość najdłuższej ścieżki od korzenia do liścia w hierarchii terminologii. Analogicznie jak w przypadku algorytmu MCA, jest to złożoność pesymistyczna.

8.4 Podsumowanie

W niniejszym rozdziale opisano praktyczne wykorzystanie zaproponowanych metod w systemie KISS oraz ocenę przedstawionych metod. Rozdział ósmy wraz z rozdziałem siódmym wykazują osiągnięcie ostatniego celu niniejszej rozprawy. Tym samym zaproponowane w rozdziałach poprzednich metody integracji wiedzy wraz z analizą ich praktycznego zastosowania dowodzą tezy rozprawy. Rozdział następny – dziewiąty – wprowadza w perspektywę dalszego rozwoju zaproponowanych metod, zaś rozdział dziesiąty podsumowuje niniejszą rozprawę.

Rozdział 9. Kierunki dalszych prac

Główny wkład niniejszej rozprawy w dziedzinę integracji wiedzy to:

- zaproponowanie nowego modelu systemu integracji wiedzy,
- zdefiniowanie źródła wiedzy i opracowanie sposobu jego tworzenia przez wzbogacenie warstwy danych o warstwę wiedzy,
- opracowanie algorytmów przetwarzania zapytań i ich walidacja w rzeczywistym systemie integracji wiedzy.

W konsekwencji, kierunki dalszych prac możemy określić jako: rozwój modelu, rozwój metod budowania źródeł wiedzy przez wzbogacanie warstwy danych o warstwę wiedzy i rozwój algorytmów przetwarzania zapytań. Szczegółowe zadania w ramach tych kierunków to:

1. Rozwój modelu systemu integracji wiedzy:
 - integracja kontekstowych źródeł wiedzy,
 - określanie tożsamości osobników w systemie integracji wiedzy,
 - wdrożenie aspektów zaufania w systemie integracji wiedzy.
2. Metody budowania źródeł wiedzy przez wzbogacenie warstwy danych o warstwę wiedzy:
 - kontekstualizacja metod budowania źródeł wiedzy,
 - budowanie źródeł wiedzy z wykorzystaniem metod ontologiczno-regułowych.
3. Algorytmy przetwarzania zapytań w systemie integracji wiedzy:
 - dostosowanie algorytmów przetwarzania zapytań do wymogów ontologiczno-regułowego systemu integracji wiedzy.
 - dostosowanie algorytmów przetwarzania zapytań do wymogów systemu integracji kontekstowych źródeł wiedzy.

W kolejnych podrozdziałach opisano te zadania.

9.1 Rozwój modelu integracji wiedzy

Integracja kontekstowych źródeł wiedzy

W rozprawie pokazano, w jaki sposób możliwe jest budowanie określonych hierarchii terminologii z wykorzystaniem mechanizmu importu. Mechanizm ten ma jednak swoje wady, zwłaszcza kiedy jest wykorzystywany do integracji wiedzy. Pierwsza z nich to budowanie złożonych ontologii importujących inne ontologie (często nie mniej złożone). W efekcie powstające w ten sposób terminologie mogą być bardzo złożone, co wydatnie zmniejsza efektywność silników wnioskujących. Druga z nich, nie mniej ważna, odnosi się do problemu

spójności ontologii, czyli istnienia informacji sprzecznych zapisanych w różnych ontologiach integrowanych. W konsekwencji zaproponowano rozwiązanie polegające na podziale ontologii na konteksty. Intuicyjnie przez kontekst rozumiemy pewien punkt widzenia na daną dziedzinę wiedzy. Przykładowo, możemy mówić o ludziach w kontekście rodziny albo w kontekście miejsca pracy (opisując dziedzinę wiedzy dotyczącą ludzi, inaczej ją opisujemy w odniesieniu do zagadnień związanych z pracą, inaczej zaś w odniesieniu do relacji rodzinnych).

W [BOUQ2005] zdefiniowano podział teorii dotyczących kontekstów na dwie grupy: *divide-and-conquer* (d-a-c), zawierającą te teorie, zgodnie z którymi podział na konteksty jest sposobem podziału ogólnej teorii świata, oraz *compose-and-conquer* (c-a-c), zawierającą te teorie, w których kontekst jest niezależną lokalną teorią, która może (ale nie musi) być powiązana z inną za pomocą specjalnych reguł wiążących.

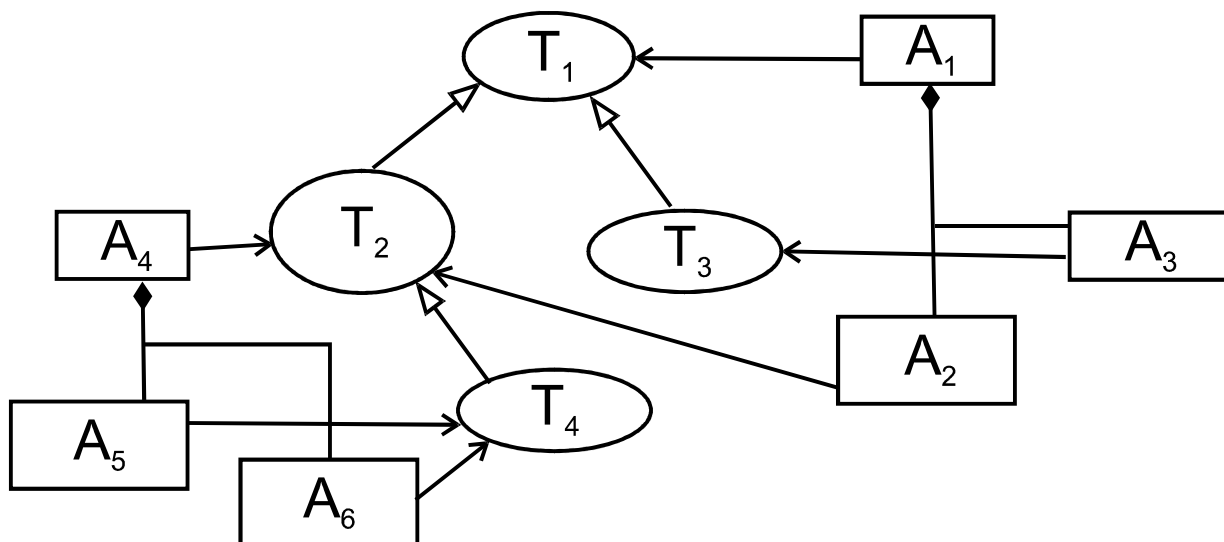
Do pierwszej grupy zaliczamy kontekstową logikę predykatów (PLC — *Propositional Logic of Context*) sformalizowaną w [BUVA1995] oraz kontekstowe uporządkowanie przestrzeni ontologicznej oparte na podejściu obiektowym. To drugie zostało zaproponowane przez A. Waloszka i przedstawione w [GOCZ2007a] i [GOCZ2007b]. Logika PLC definiuje sposób tworzenia kontekstu i wnioskowania w jego ramach. Każda formuła musi być zlokalizowana w jakimś kontekście i tylko wtedy jest sens mówić o tym, czy jest prawdziwa, czy nie. Model PLC definiuje funkcję zwaną słownikiem, która wiąże formułę mającą znaczenie w danym kontekście z tym kontekstem. Konteksty są uporządkowane hierarchicznie. Wznoszenie się ponad i wykraczanie poza kontekst są metodami wnioskowania o zdaniach należących do jednego kontekstu z punktu widzenia innego kontekstu.

Kontekstualizacja przestrzeni ontologicznej oparta na podejściu obiektowym buduje hierarchię conceptów, wykorzystując przy tym relację dziedziczenia kontekstów. Dziedziczenie kontekstów dotyczy tylko części terminologicznej ontologii. Podstawowym prawem jest to, że terminologia bardziej wyspecjalizowana – kontekst podrzędny – „widzi” terminologię bardziej ogólną – kontekst nadrzędny – ale nie odwrotnie. Kontekst nadrzędny nawet „nie wie” o istnieniu kontekstów podrzędnych. Uporządkowanie opisu świata jest dokonane przez możliwość tworzenia wielu opisów świata dla jednego kontekstu – wystąpień kontekstów. Jeśli powiemy, że konteksty są odpowiednikami klas w paradygmacie obiektowym, to możemy powiedzieć, że opisy świata są odpowiednikami wystąpień klas.

Podstawę dla podejścia c-a-c tworzy semantyka modelu lokalnego/systemu wielokontekstowego (LMS/MCS — *Local Model Semantics/Multi-context System*), opublikowana w [GHID2001] i [GIUN1994]. Większość prac bazujących na tym podejściu wykorzystuje reguły pomostowe (ang. *bridge rules*) [GIUN1993], które są opisem zależności między dwoma zbiorami informacji.

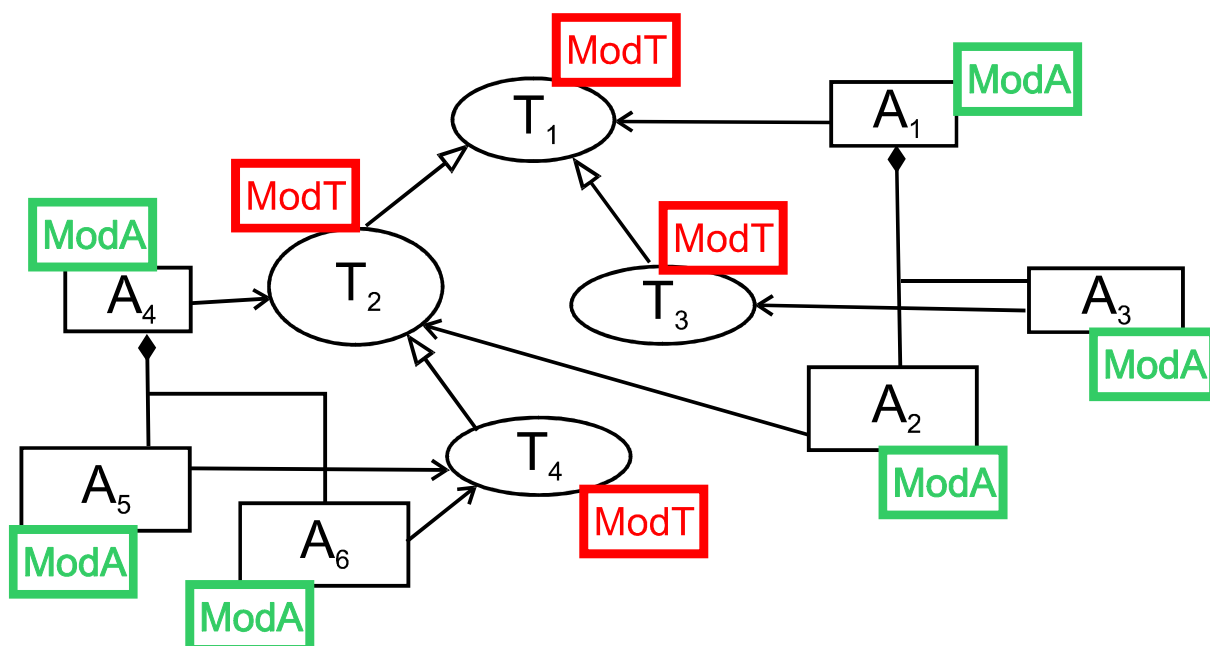
Z rozwojem systemu integracji wiedzy wiąże się integracja kontekstowych źródeł wiedzy. Na rysunku 9.1 przedstawiono przykładową strukturę ontologii kontekstowej. T_1, \dots, T_4 to konteksty, zaś A_1, \dots, A_6 to wystąpienia kontekstów. Zwykłe strzałki pomiędzy wystąpieniem kontekstu, a kontekstem oznaczają zależność pomiędzy kontekstem a jego wystąpieniem (np.: A_4 jest wystąpieniem kontekstu T_2). Strzałki dziedziczenia oznaczają hierarchię dziedziczenia kontekstów, zaś strzałki agregacji hierarchię agregacji wystąpień kontekstów. W zaproponowanym w tej rozprawie modelu systemu integracji wiedzy zdefiniowano źródło wiedzy jako system reprezentacji wiedzy. Źródło wiedzy z założenia dostarcza mechanizmów wnioskowania dla terminologii TBox i zadanego opisu świata ABox. Analogię dla źródła wiedzy w systemie kontekstowym stanowi **kontekstowe źródło wiedzy**. Kontekstowe źródło wiedzy składa się z dwóch rodzajów modułów: *ModT* i *ModA*. Każdy moduł *ModT* odpowiada za jeden kontekst – w omawianym przykładzie istnieją cztery wystąpienia modułów *ModT*: dla kontekstów T_1, T_2, T_3 i T_4 . Funkcjonalność modułu *ModT*

polega na wnioskowaniu z terminologii. Podczas wnioskowania jeden moduł porozumiewa się z modułami *ModT* innych kontekstów. Wnioskowanie z jednego kontekstu angażuje moduł *ModT* tego kontekstu oraz moduły kontekstów dziedziczonych. Zakłada się, że w przyszłości, na dalszym etapie rozwoju modelu, możliwe też będzie wnioskowanie z wielu kontekstów, które obejmie swoim zasięgiem większą liczbę modułów.



Rysunek 9.1. Ontologia kontekstowa

Każdy moduł *ModA* jest odpowiedzialny za jedno wystąpienie kontekstu. W powyższym przykładzie istnieje sześć wystąpień modułu *ModA*: dla wystąpień kontekstów A_1, \dots, A_6 . Podczas wnioskowania moduł *ModA* porozumiewa się z modułem *ModT* odpowiedniego kontekstu oraz z modułami *ModA* wystąpień agregowanych i agregujących. Moduł *ModT* wspiera proces wnioskowania informacjami terminologicznymi. Komunikacja z modułami *ModA* ma na celu zebranie wniosków od wystąpień, które znajdują się poniżej i powyżej danego wystąpienia kontekstu w hierarchii narzuconej przez relację agregacji. *ModA* łączy te wnioski i zwraca podmiotowi zadającemu zapytanie. W przedstawionym w tej pracy systemie integracji wiedzy zostały zdefiniowane dwa typy źródeł wiedzy: baza wiedzy (przykładową opisaną bazą wiedzy jest system KaSeA) i system wnioskowania z zewnętrznych źródeł danych (przykładowym opisanym systemem jest system KL). W systemie integracji kontekstowych źródeł wiedzy nie istnieją systemy tożsame z systemem KaSeA lub KL. Ich funkcjonalność zostałaby rozdzielona na moduły *ModT* i *ModA*., przy czym funkcjonalność modułu *ModT* jest taka sama dla obu systemów, zaś funkcjonalność modułu *ModA* dla systemu KL i KaSeA jest różna. Moduł *ModA* dla systemu KaSeA dostarcza metod wnioskowania z opisu świata zapisanego w bazie wiedzy, zaś moduł *ModA* dla systemu KL dostarcza metod wnioskowania z niejawnie zapisanego opisu świata w zewnętrznych źródłach danych. W systemie integracji dodatkowo istnieje jeszcze jeden typ modułu *ModA* odpowiedzialny za integrowanie różnych opisów świata (jawnych i niejawnych) pochodzących z różnych źródeł danych i baz wiedzy dla jednego wystąpienia kontekstu. Moduły *ModA* i *ModT* dla przykładowej ontologii kontekstowej zostały przedstawione na rysunku 9.2.



Rysunek 9.2. Moduły *ModA* i *ModT* dla przykładowej ontologii kontekstowej

W systemie integracji wiedzy pochodzącej z kontekstowych źródeł wiedzy należałoby również zdefiniować sposób odwzorowywania terminów z jednej terminologii kontekstowej w inną. Taki sposób odwzorowania bazowałby, tak jak w przypadku opisanego w pracy systemu integracji, na pajęczynie. Pajęczyny takie musiałyby jednak uwzględniać kontekstowy podział terminologii.

Określanie tożsamości osobników w systemie integracji wiedzy

W rozdziale 2 został postawiony problem identyfikowalności zasobów w Sieci Semantycznej. W Sieci Semantycznej zasoby są identyfikowane poprzez URI. Jednak ze względu na fakt, że każdy użytkownik Sieci Semantycznej może nadawać URI niezależnie od innych użytkowników, te same semantycznie zasoby mogą mieć różne identyfikatory URI. W niniejszej analizie koncentrujemy się na problemie znajdowania osobników o tej samej tożsamości. Problem semantycznie tożsamyh konceptów, ról i atrybutów jest zagadnieniem dotyczącym integracji terminologii i w zaproponowanym w rozprawie modelu systemu integracji wiedzy jest rozwiązywany przez człowieka na etapie budowania odwzorowań aksjomatycznych. Aktualnie w Sieci Semantycznej tożsamość osobników może być definiowana przy użyciu pewnych konstrukcji języka OWL [OWL2004] i reguł SWRL [SWRL2004].

W języku OWL nie ma założenia o unikatowości nazw. Istnieją trzy konstrukcje, które mogą być używane do określenia tożsamości osobników:

- 1) `owl:sameAs` używana w celu wskazania, że dwa identyfikatory URI wskazują na tego samego osobnika,
- 2) `owl:differentFrom` używana w celu wskazania, że dwa identyfikatory URI na pewno nie wskazują na tego samego osobnika,
- 3) `owl:AllDifferent` używana w celu wskazania, że wszystkie identyfikatory URI znajdujące się na liście wskazują na różnych osobników.

Za pomocą tych konstrukcji możemy jawnie wskazać, że dwa identyfikatory URI wskazują na tego samego osobnika lub na dwa różne osobniki. Informacja ta nie jest

wywiedziona. W języku OWL istnieje jeszcze jedna konstrukcja, która może być wykorzystana do określenia, że dwa identyfikatory URI wskazują na ten sam lub dwa różne osobniki. Jest to konstrukcja `owl:InverseFunctionalProperty`. W języku OWL w ten sposób może być oznaczona zarówno rola, jak i atrybut (oznaczenie atrybutu tym znacznikiem wykracza jednak poza OWL-DL). Jeśli rola R oznaczona jest znacznikiem `owl:InverseFunctionalProperty`, oznacza to, że ten sam osobnik nie może być dopełnieniem roli R dla dwóch różnych osobników:

$$R(x_1, y) \wedge R(x_2, y) \rightarrow x_1 = x_2$$

Analogiczna sytuacja zachodzi w przypadku atrybutów. Atrybut oznaczony jako `owl:InverseFunctionalProperty` może być traktowany analogicznie jak wartość unikatowa w bazach danych.

Konstrukcja `owl:InverseFunctionalProperty` umożliwia identyfikowanie tożsamości osobników na podstawie wartości atrybutu („klucza”) lub osobnika będącego dopełnieniem roli.

Przykład 9.1 Zastosowanie konstrukcji `owl:InverseFunctionalProperty` do identyfikowania tożsamości osobników

```
<owl:InverseFunctionalProperty rdf:ID="PESEL">
  <rdfs:domain rdf:resource="#Człowiek"/>
  <rdfs:range rdf:resource=" &xsd:string"/>
</owl:InverseFunctionalProperty>
```

Przykład pokazuje, że człowiek jest jednoznacznie identyfikowany wartością atrybutu PESEL.

Za pomocą konstrukcji `owl:InverseFunctionalProperty` nie można zamodelować bardziej złożonych reguł określających tożsamość osobników. Przykładowo, jeśli chcielibyśmy wyrazić, że dwa identyfikatory URI wskazują na to samo mieszkanie, jeśli każde z tych mieszkań znajduje się w tym samym mieście, przy tej samej ulicy i ma ten sam numer domu, to jest to niemożliwe. Również nowa wersja języka OWL, OWL 1.1 [OWL1.12006], nie dostarcza nowych konstrukcji umożliwiających modelowanie bardziej złożonych reguł określających tożsamość osobników. Możliwość taką dostarcza jednak język SWRL: *A Semantic Web Rule Language*.

SWRL rozszerza konstrukcje języka OWL o klauzule Horna. Klauzule Horna to reguły postaci:

$$A_1, \dots, A_n \rightarrow B$$

gdzie A_i i B są atomowymi formułami logicznymi. Znaczenie takiej reguły jest następujące: Jeśli wszystkie A_i są prawdziwe, to także B jest prawdziwe. W szczególnym przypadku wszystkie formuły A_i mogą być puste, co oznacza bezwarunkową prawdziwość B . W języku SWRL atomową formułą logiczną mogą być formuły postaci $C(x)$, $P(x, y)$, `sameAs(x, y)`, `differentFrom(x, y)` oraz `builtIn(r, z1, ..., zn)`, gdzie C oznacza koncept, P rolę lub atrybut, a x, y, z_i zmienne, osobniki lub wartości. Formuła postaci `builtIn(r, z1, ..., zn)` oznacza, że zmienne z_1, \dots, z_n pozostają ze sobą w relacji r , co zobrazowano w następującym przykładzie:

Przykład 9.2 Przykład formuły `builtIn`

Możemy wykorzystać formułę `builtIn` między innymi do porównywania dwóch wartości typu *string*:

```
builtIn(swrlb:stringEqualIgnoreCase, z1, z2)
```

Za pomocą reguł Horna możliwe jest modelowanie złożonych zależności, również określających tożsamość osobników. I tak, w łatwy sposób możemy wyrazić, że dwa identyfikatory URI wskazują na to samo mieszkanie, jeśli każde z mieszkań znajduje się w tym samym mieście, przy tej samej ulicy i ma ten sam numer domu.

Przykład 9.3 Zastosowanie reguł Horna do identyfikowania tożsamości osobników

Zamodelujmy wycinek terminologii dotyczący mieszkań.

$$\text{Mieszkanie} \sqsubseteq \top \quad (1)$$

$$\text{Miasto} \sqsubseteq \top \quad (2)$$

$$\text{Ulica} \sqsubseteq \top \quad (3)$$

$$\text{Mieszkanie} \sqcap \text{Miasto} \equiv \perp \quad (4)$$

$$\text{Mieszkanie} \sqcap \text{Ulica} \equiv \perp \quad (5)$$

$$\text{Miasto} \sqcap \text{Ulica} \equiv \perp \quad (6)$$

$$\text{Mieszkanie} \sqsubseteq \exists \text{jestWMieście.Miasto} \quad (7)$$

$$\text{Mieszkanie} \sqsubseteq \exists \text{jestNaUlicy.Ulica} \quad (8)$$

$$\text{Mieszkanie} \sqsubseteq \exists \text{maNrDomu.string} \quad (9)$$

Każde mieszkanie znajduje się w pewnym mieście i na pewnej ulicy. Również każde mieszkanie ma swój numer domu. Chcemy teraz, używając reguł Horna, określić, że dwa identyfikatory URI wskazują na to samo mieszkanie, jeśli każde z mieszkań znajduje się w tym samym mieście, przy tej samej ulicy i ma ten sam numer domu. Taka reguła wygląda następująco:

$$\begin{aligned} &\text{Mieszkanie}(a), \text{Mieszkanie}(b), \text{jestWMieście}(a, c), \text{jestWMieście}(b, c), \\ &\text{jestNaUlicy}(a, d), \text{jestNaUlicy}(b, d), \text{maNrDomu}(a, e), \text{maNrDomu}(b, e) \rightarrow \text{sameAs}(a, b) \end{aligned}$$

Język SWRL jest językiem ogólnym, daleko wykraczającym poza zakres reguł potrzebnych do identyfikowania tożsamości osobników. Jednak do identyfikacji osobników potrzebne byłyby również dodatkowe formuły, niemożliwe do wyrażenia w języku SWRL. Przykładowo, w języku SWRL nie możemy wyrazić, że dwa identyfikatory URI wskazują na te same przepisy, jeśli składniki tych przepisów są takie same. Dlatego w zespole KMG@GUT prowadzone są prace nad metodą UII (*Unique Individual Identification*), która dostarcza zbioru reguł identyfikacji osobników. Zbiór ten jest pewnym podzbiorem zbioru reguł Horna obsługiwanych przez język SWRL, wzbogaconych o specyficzne formuły potrzebne do identyfikacji osobników. Zbiór reguł i formuł w metodzie UII został wybrany w taki sposób, aby metoda UII wykazywała następujące cechy umożliwiające jej praktyczne wykorzystanie w Sieci Semantycznej:

- 1) reguły mają ekspresywność wystarczającą do identyfikacji osobników w Sieci Semantycznej,
- 2) przetwarzanie reguł użytych w metodzie UII przez system wnioskujący nie zwiększa zasadniczo złożoności algorytmów wnioskujących, zarówno w środowisku lokalnym, jak i rozproszonym.

W metodzie UII, tak jak w przypadku języka SWRL, reguły mają postać klauzul Horna. Przyjmijmy, że reguły definiowane w metodzie UII będziemy dalej nazywać regułami identyfikowalności osobników. Formuła logiczna występująca jako następnik w regule identyfikowalności osobników jest zawsze formułą postaci $\text{sameAs}(x, y)$. Formuły logiczne mogące występować jako poprzedniki to:

- 1) $C(x)$,
- 2) $P(x, y)$,
- 3) $\text{builtIn}(r, z_1, \dots, z_n)$
- 4) $\text{allKnown}(P, x, y)$,

gdzie C oznacza koncept, P rolę lub atrybut, a x, y zmienne, osobniki lub wartości. Każda formuła logiczna przyjmuje wartość prawdziwą lub fałszywą: prawdziwą, jeśli wiemy, że na pewno zachodzi dana formuła, zaś fałszywą w przeciwnym wypadku. Znaczenie formuł logicznych $C(x)$, $P(x, y)$, $\text{builtIn}(r, z_1, \dots, z_n)$, $\text{sameAs}(x, y)$ jest analogiczne jak w języku SWRL. Znaczenie formuły $\text{allKnown}(P, x, y)$ zostało przedstawione w definicji 9.1.

Definicja 9.1 Definicja formuły logicznej $\text{allKnown}(P, x, y)$

Niech \mathcal{I} oznacza interpretację będącą modelem pewnej terminologii \mathcal{T} . Znaczenie formuły $\text{allKnown}(P, x, y)$ jest następujące:

$$(\text{allKnown}(P, x, y) = \text{true}) \leftrightarrow (\forall z (z^{\mathcal{I}} \in \Delta^{\mathcal{I}}) (\mathbf{K}P(x, z) \leftrightarrow \mathbf{K}P(y, z)))$$

Formuła $\text{allKnown}(P, x, y)$ jest prawdziwa wtedy i tylko wtedy, gdy:

- 1) jeśli P jest atrybutem: to zbiór wszystkie znanych przez bazę wiedzy wartości atrybutu P dla osobnika x jest równy zbiorowi wszystkich znanych przez bazę wiedzy wartości atrybutu P dla osobnika y ,
- 2) jeśli P jest rolą: to zbiór wszystkich znanych przez bazę wiedzy dopełnień roli R dla osobnika x jest równy zbiorowi wszystkich dopełnień roli R dla osobnika y .

Za pomocą formuły $\text{allKnown}(P, x, y)$ możliwe jest wyrażenie, że dwa przepisy są równe, jeśli mają takie same składniki:

$$\text{sameAs}(x, y) \leftarrow \text{Przepis}(x) \wedge \text{Przepis}(y) \wedge \text{allKnown}(\text{maSkładnik}, x, y)$$

W definicji 9.1 został użyty operator \mathbf{K} nazywany *operatorem epistemicznym* (ang. *epistemic operator*) [DONI1992], [DONI1998]. Operator \mathbf{K} oznacza, że: ‘baza wiedzy wie, że...’. W przypadku formuły $\text{allKnown}(P, x, y)$ użycie operatora \mathbf{K} jest jak najbardziej uzasadnione. Wynika to z przyjęcia założenia o świecie otwartym. Jeśli nie użylibyśmy operatora \mathbf{K} , oznaczałoby to, że wszystkie składniki danego przepisu mają takie same składniki. Jednak baza wiedzy nie wie, czy poza składnikami, o których „wie”, nie istnieją jeszcze inne składniki, a tym samym nie może stwierdzić, czy te inne „nie znane” jej składniki są takie same dla obu przepisów.

Przedstawiona idea odejścia od założenia unikatowości nazw osobników (UNA) jest aktualnie w początkowym stadium rozwoju. Jakkolwiek wzbogacenie języka SWRL o nową formułę postaci $\text{allKnown}(P, x, y)$ wynikało z analizy rzeczywistych źródeł wiedzy dotyczących odżywiania zastosowanych w projekcie PIPS. Metoda UII zakłada rozszerzalność formuł logicznych w zależności od potrzeb, przy założeniu możliwości stworzenia efektywnych algorytmów wnioskujących.

Wdrożenie aspektów zaufania w systemie integracji wiedzy

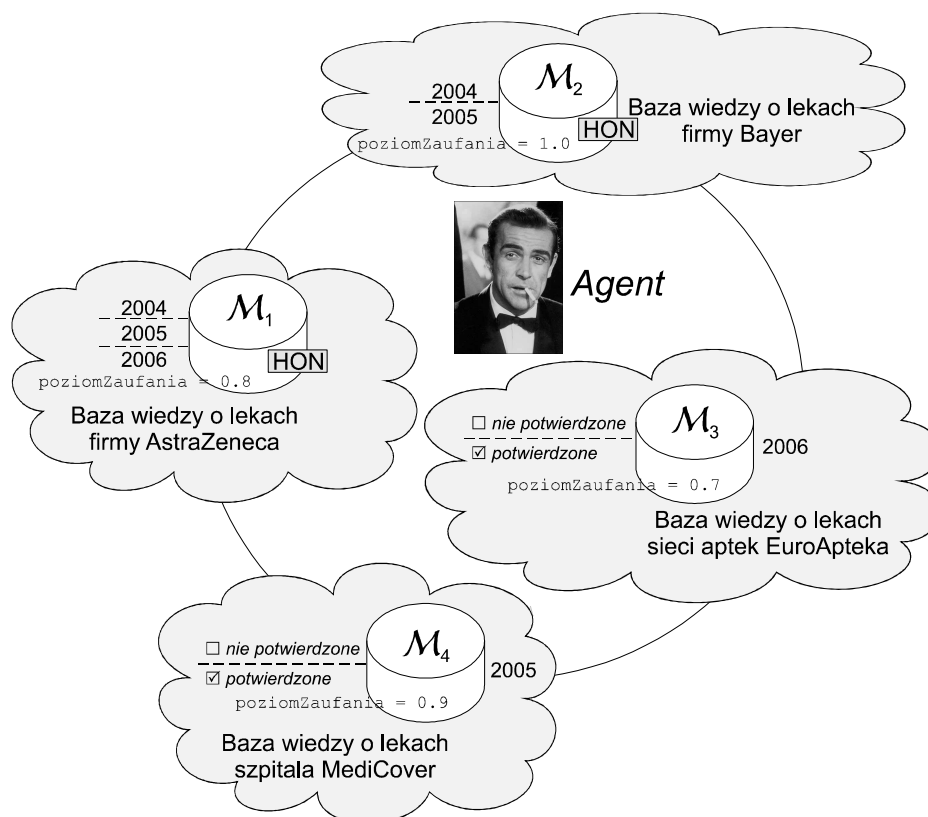
W pracach [ZAWA2007], [GOCZ2008b], [GOCZ2008c] zaproponowano system zarządzania wiedzą z uwzględnieniem aspektów zaufania. Kierunki dalszych prac obejmują również połączenie przedstawionego w pracy modelu systemu integracji wiedzy z modelem systemu zarządzania wiedzą zaproponowanym w [GOCZ2008b].

Definiowanie wiarygodności w systemie zarządzania wiedzą zaproponowanym w [ZAWA2007] jest rozumiane jako przypisanie asercjom unarnym i binarnym zbioru cech i ich wartości. W ten sposób asercje są reifikowane, a wnioskowanie z uwzględnieniem aspektów wiarygodności polega na utworzeniu dynamicznego modelu ontologii zawierającego tylko te asercje, których cechy spełniają zadane wymagania, a następnie wykonanie zapytania dziedzinowego na tym właśnie modelu.

Posłużmy się kilkoma przykładami zapytań skierowanych do agentów w Sieci Semantycznej.

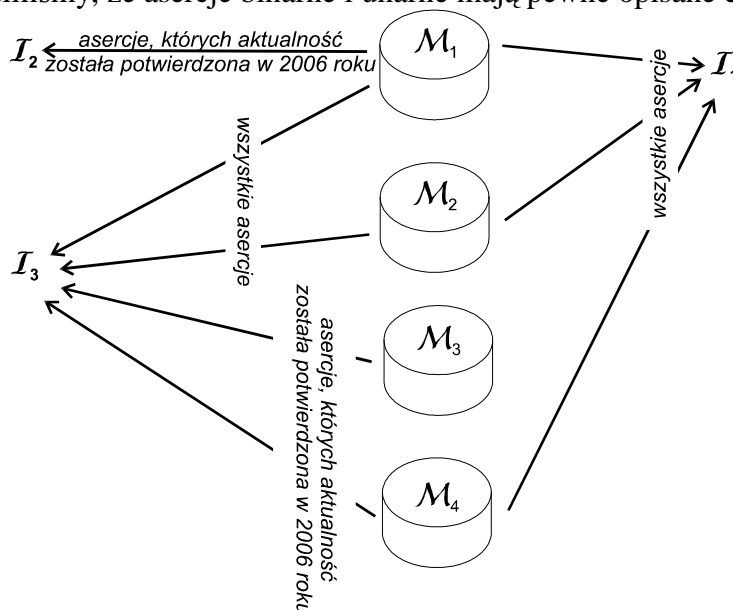
1. „Podaj mi informacje o lekach, ale tylko takie, które pochodzą ze źródeł o stopniu zaufania 0,8 lub wyższym”.
2. „Podaj mi informacje o lekach, ale tylko takie, które posiadają certyfikat HON (*Health on the Net Foundation*) [HON2008] i ich aktualność została potwierdzona w 2006 roku”.
3. „Podaj mi informacje o lekach, ale tylko takie, które potwierdził mój lekarz rodzinny, lub takie, które mają certyfikat HON”.

Wszystkie te zadania, z punktu widzenia agenta w Sieci Semantycznej, sprowadzają się do zbudowania odpowiedniego modelu dla ontologii leków. Dla każdego z trzech powyższych zapytań agent koncepcyjnie budowałby model danej ontologii. Rysunek 9.2 przedstawia przykładowe bazy wiedzy o lekach w Sieci Semantycznej. Wszystkie te bazy wiedzy są opisane pewną terminologią leków \mathcal{T} . Każda z baz wiedzy \mathcal{M}_1 , \mathcal{M}_2 , \mathcal{M}_3 i \mathcal{M}_4 utworzona jest w wyniku wczytania do ontologii zbioru asercji binarnych i unarnych. Asercje te są charakteryzowane dodatkowo przez pewne cechy. Przykładowo: wiadomo, że do bazy wiedzy o lekach firmy AstraZeneca zostały wprowadzone asercje potwierdzone przez HON oraz aktualność tych asercji została potwierdzona w roku 2004, 2005 lub 2006. W przypadku, kiedy agent otrzyma zapytanie z opisem wiarygodności, jaki musi spełniać dostarczona przez niego wiedza, tworzy dynamiczny model, w którym każdemu conceptowi i każdej roli przyporządkowane są takie relacje binarne i unarne, które spełniają określone warunki. Na rysunku 9.3 pokazane są te relacje, które znajdują się w tak utworzonym modelu dynamicznym dla poszczególnych zapytań. I tak, przykładowo, model pierwszy jest to taka interpretacja \mathcal{I}_1 , która przypisuje conceptom i rolom odpowiednio te relacje unarne i binarne, które mają stopień zaufania 0,8 lub wyższy. Interpretacja \mathcal{I}_1 zostaje utworzona w wyniku wczytania asercji binarnych i unarnych z baz wiedzy \mathcal{M}_1 , \mathcal{M}_2 i \mathcal{M}_4 . Zauważmy, że interpretacja \mathcal{I}_1 będzie modelem wtedy i tylko wtedy, gdy asercje pobrane z baz wiedzy nie są wzajemnie sprzeczne. Zachowanie systemu w przypadku sprzeczności danych przy budowaniu modelu dynamicznego może się różnić w zależności od poszczególnych implementacji – można na przykład założyć, że takiego modelu nie da się zbudować lub że usuwane są te asercje, które powodują, że interpretacja przestaje być modelem.



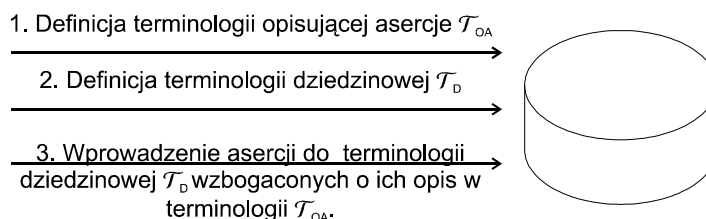
Rysunek 9.3. Przykładowe źródła wiedzy o lekach w Sieci Semantycznej

Wcześniej określiliśmy, że asercje binarne i unarne mają pewne opisane cechy.



Rysunek 9.4. Dynamiczne budowanie interpretacji w Sieci Semantycznej

Wartości tych cech mogą być takie same lub różne dla różnych asercji. Agent, aby odpowiedzieć na dane zapytanie na pewnym poziomie wiarygodności, musi zapytać każdą z ontologii opisujących bazy wiedzy o zbiór asercji unarnych i binarnych spełniających określone warunki, czyli o asercje posiadające pewne określone wartości dla poszczególnych cech. Proces tworzenia opisu ontologicznego asercji został przedstawiony na rysunku 9.4.



Rysunek 9.5. Proces tworzenia opisu ontologicznego asercji

Proces ten składa się z trzech podstawowych etapów. W etapie pierwszym definiowana jest terminologia umożliwiająca opisanie wiarygodności asercji \mathcal{T}_{OA} . Drugi etap to zdefiniowanie terminologii dziedzinowej \mathcal{T}_D , zaś etap trzeci to wstawianie asercji do obu terminologii, jednak w taki sposób, że asercje z terminologii \mathcal{T}_D stają się osobnikami w terminologii \mathcal{T}_{OA} .

W systemie zarządzania wiedzą nie w pełni wiarygodną dynamicznie budowany jest opis świata dla terminologii dziedzinowej. System integracji wiedzy nie w pełni wiarygodnej powinien umożliwiać dynamiczne budowanie nie tylko opisu świata, ale także terminologii (różne źródła wiedzy mogą być opisane różnymi terminologiami opisującymi tę samą dziedzinę wiedzy). System taki zbierałby asercje z różnych źródeł wiedzy, opisanych różnymi terminologiami, a nie, tak jak w przypadku istniejącego systemu, tylko ze źródeł wiedzy opisanych jednolitą terminologią dziedzinową. W systemie integracji wiedzy nie w pełni wiarygodnej, tak jak w systemie integracji wiedzy zaproponowanym w niniejszej rozprawie, zapisany byłby zbiór odwzorowań aksjomatycznych. Odwzorowania te byłyby wykorzystywane do dynamicznego budowania terminologii, w terminach której zostało wyrażone zapytanie. System, zbierając asercje z różnych źródeł wiedzy, przekształcałby je w asercje wyrażone w terminach terminologii dynamicznie utworzonej. W taki sposób uzyskalibyśmy system integracji wiedzy nie w pełni wiarygodnej.

9.2 Metody budowania źródeł wiedzy przez wzbogacenie warstwy danych o warstwę wiedzy

Kontekstualizacja metod budowania źródeł wiedzy

Zaproponowane w rozprawie metody SED i RED umożliwiają nie tylko opisywanie danych za pomocą terminów ontologicznych, ale również wnioskowanie z tych danych analogicznie jak z opisu świata zdefiniowanego w ontologii. Metoda SED, wzbogacając dane o opis ich semantyki, umożliwia tworzenie źródeł wiedzy składających się z terminologii i dynamicznie tworzonego opisu świata. Zaproponowany w tej rozprawie system nie umożliwia jednak przetwarzania danych opisanych ontologią kontekstową. Wzbogacenie metod SED i RED o możliwości wnioskowania kontekstowego jest naturalnym krokiem rozwoju opisanych metod umożliwiającym ich jeszcze szersze praktyczne zastosowanie w systemach integracji danych.

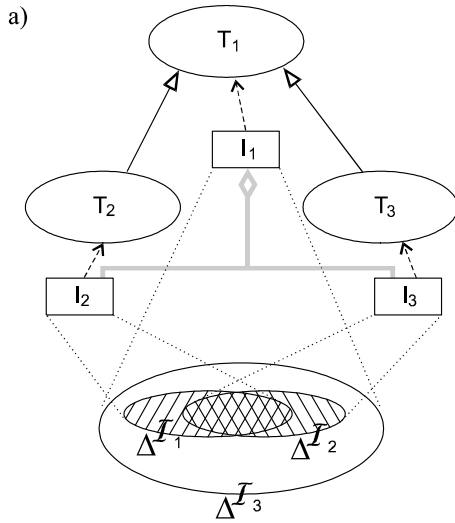
Do rozwoju metod SED i RED zastosowano kontekstualizację przestrzeni ontologicznej opartej na podejściu obiektowym. Za taką decyzję przemawiają poniższe przesłanki. Kontekstualizacja przestrzeni ontologicznej oparta na podejściu obiektowym umożliwia:

- opisywanie różnych fragmentów wiedzy na różnych poziomach szczegółowości (tzn. przy użyciu terminów o różnym poziomie ogólności),
- wprowadzenie hierarchicznego porządku w dużych ontologiach,

- przechowywanie sprzecznych asercji, jeśli przedstawiają one ten sam problem z różnych punktów widzenia,
- zintegrowanie informacji widzianych z różnych punktów widzenia na żądanym poziomie uogólnienia.

Ponadto system KaSeA wspiera ten typ kontekstualizacji, reprezentując go przy wykorzystaniu sygnatur zdefiniowanych w kartograficznej reprezentacji wiedzy.

W metodzie SED definiujemy odwzorowania dla ról, atrybutów i konceptów. Dostosowanie metody SED do kontekstowego opisu danych wymaga zdefiniowania odwzorowań dla ról, atrybutów i konceptów w odniesieniu do zadanego kontekstu i wystąpienia kontekstu. Stosowne definicje przytoczono poniżej.



- b)
- T_1 $Człowiek \equiv \exists maAdres.string$
 $Człowiek \equiv \exists maPESEL.string$
 $Człowiek \equiv \exists maPłeć.Płeć$
 $Płeć \equiv \{żeńska, męska\}$
 $Kobieta \equiv \exists maPłeć.\{żeńska\}$
 $Mężczyzna \equiv \exists maPłeć.\{męska\}$
- T_2 $\exists maStanowisko.Stanowisko \sqsubseteq Człowiek$
 $Stanowisko \sqcap Człowiek \equiv \perp$
 $Pracownik \sqsubseteq \exists maStanowisko.Stanowisko$
- T_3 $Rodzic \equiv \exists maDziecko.Człowiek$
 $\exists maDziecko.\neg Człowiek \sqsubseteq \perp$
 $Kobieta \sqcap Rodzic \equiv Matka$
 $Mężczyzna \sqcap Rodzic \equiv Ojciec$

Rysunek 9.6. a) Hierarchia kontekstów [GOCZ2007a] i b) przykładowe aksjomaty zdefiniowane w kontekstach terminologii O_x .

Definicja 9.2 Odwzorowanie konceptu $M_{C_x, I, C} = \langle C_x, I, C, q_{C_x, I, C} \rangle$

Odwzorowaniem konceptu $M_{C_x, I, C} = \langle C_x, I, C, q_{C_x, I, C} \rangle$ nazywamy czwórkę składającą się z kontekstu C_x , wystąpienia kontekstu I , konceptu C , atomowego lub złożonego, wyrażonego w logice opisowej, oraz zapytania $q_{C_x, I, C}$ do źródła danych, dla którego odpowiedzią $r_{C_x, I, C} = \{i: i \text{ jest nazwą osobnika}\}$ jest zbiór nazw osobników.

Definicja 9.3 Odwzorowanie roli $M_{C_x, I, R} = \langle C_x, I, R, q_{C_x, I, R} \rangle$

Odwzorowaniem roli $M_{C_x, I, R} = \langle C_x, I, R, q_{C_x, I, R} \rangle$ nazywamy czwórkę składającą się z kontekstu C_x , wystąpienia kontekstu I , roli R , atomowej lub złożonej, wyrażonej w logice opisowej oraz zapytania $q_{C_x, I, R}$ do źródła danych, dla którego odpowiedzią

$$r_{C_x, I, R} = \{ \langle i, j \rangle : i, j \text{ są nazwami osobników} \}$$

jest zbiór par nazw osobników.

Definicja 9.4 Odwzorowanie atrybutu $M_{C_x, I, A} = \langle C_x, I, A, q_{C_x, I, A} \rangle$

Odwzorowaniem atrybutu $M_{C_x, I, A} = \langle C_x, I, A, q_{C_x, I, A} \rangle$ nazywamy czwórkę składającą się z kontekstu C_x , wystąpienia kontekstu I , atrybutu A wyrażonego w logice opisowej oraz zapytania $q_{C_x, I, A}$ do źródła danych, dla którego odpowiedzią

$$r_{C_x, I, A} = \{ \langle i, v \rangle : i \text{ jest nazwą osobnika, a } v \text{ pewną wartością} \}$$

jest zbiór par: nazwa osobnika, wartość.

Definicja 9.5 Zbiór odwzorowań S

Zbiór odwzorowań $S = S_{M_{C_x, I, C}} \cup S_{M_{C_x, I, R}} \cup S_{M_{C_x, I, A}}$ definiujemy jako sumę zbiorów: odwzorowań konceptów $S_{M_{C_x, I, C}}$, odwzorowań ról $S_{M_{C_x, I, R}}$ i odwzorowań atrybutów $S_{M_{C_x, I, A}}$.

W celu zobrazowania znaczenia odwzorowań dla ontologii kontekstowych posłużmy się przykładem. Na rysunku 9.6 zdefiniowane są trzy konteksty: T_1 , T_2 i T_3 . W kontekście T_1 zapisane są pewne ogólne stwierdzenia dotyczące ludzi, mówiące o tym, że każdy człowiek ma pewien adres zamieszkania (zakładamy, że jest to adres zameldowania i nie ma ludzi bezdomnych). Ponadto każdy człowiek ma swój PESEL i jest albo mężczyzną, albo kobietą: mężczyzną, jeśli jest płci męskiej, kobietą, jeśli jest płci żeńskiej. W kontekście T_2 zapisane są aksjomaty mówiące o tym, że jeśli ktoś ma jakieś stanowisko, to musi być pracownikiem. W kontekście T_3 zapisane są aksjomaty dotyczące relacji ojcostwa i macierzyństwa. Możemy więc stwierdzić, że kontekst T_1 dotyczy danych formalnych człowieka, kontekst T_2 informacji związanych z zatrudnieniem, zaś kontekst T_3 informacji związanych z rodzicielstwem. Oczywiście wszystkie trzy terminologie są bardzo uproszczone dla celów przykładu.

Na rysunku 9.6 widzimy również trzy wystąpienia kontekstów: l_1 , l_2 i l_3 , połączone relacją agregacji. Zdefiniujmy przykładowy zbiór odwzorowań S dla pewnego źródła relacyjnego Z .

1) Zbiór odwzorowań konceptów $S_{M_{C_x, I, C}}$:

a. $M_{T_1, l_1, Kobieta} = \langle T_1, l_1, Kobieta, q_{T_1, l_1, Kobieta} \rangle$, gdzie

$$q_{T_1, l_1, Kobieta} = \text{select PESEL from LUDZIE where PLEC} = \text{'k'} \text{ and} \\ \text{PELNOLETNI} = \text{'true'}$$

b. $M_{T_1, l_1, Meczczyna} = \langle T_1, l_1, Meczczyna, q_{T_1, l_1, Meczczyna} \rangle$, gdzie

$$q_{T_1, l_1, Meczczyna} = \text{select PESEL from LUDZIE where PLEC} = \text{'m'} \text{ and} \\ \text{PELNOLETNI} = \text{'true'}$$

2) Zbiór odwzorowań ról $S_{M_{C_x, I, R}}$:

a. $M_{T_2, l_2, maStanowisko} = \langle T_2, l_2, maStanowisko, q_{T_2, l_2, maStanowisko} \rangle$, gdzie

$$q_{T_2, l_2, maStanowisko} = \text{select PESEL, NAZWA_STANOWISKA from LUDZIE,} \\ \text{STANOWISKA where} \\ \text{LUDZIE.IDSTANOWISKA} = \text{STANOWISKA.IDSTANOWISKA}$$

b. $M_{T_3, l_3, maDziecko} = \langle T_3, l_3, maDziecko, q_{T_3, l_3, maDziecko} \rangle$, gdzie

$$q_{T_3, l_3, maDziecko} = \text{select R.PESEL, DZ.PESEL from LUDZIE R,}$$

LUDZIE DZ, DZIECI where DZIECI.RODZIC = R.PESEL
and DZIECI.DZIECKO = DZ.PESEL

3) Zbiór odwzorowań atrybutów $S_{M_{C_x, I, A}}$:

- a. $M_{T_1, I_1, maAdres} = \langle T_1, I_1, maAdres, q_{T_1, I_1, maAdres} \rangle$, gdzie
 $q_{T_1, I_1, maAdres} = \text{select PESEL, ADRES from LUDZIE}$

Po zdefiniowaniu zbioru odwzorowań możliwe jest utworzenie ontologii $O[O_x, S]$.

Definicja 9.6 Ontologia $O[O_x, S]$

Ontologią $O[O_x, S]$ dla zbioru odwzorowań S nazywamy parę $\langle O_x, O_s[S] \rangle$ składającą się z terminologii O_x oraz zbioru asercji $O_s[S]$ utworzonych na podstawie zbioru odwzorowań S w taki sposób, że:

$$O_s[S] = \{ \langle C_x, I, C(i) \rangle : \exists M_{C_x, I, C} \in S_{M_{C_x, I, A}} (i \in r_{C_x, I, C}) \} \cup \\ \{ \langle C_x, I, R(i, j) \rangle : \exists M_{C_x, I, R} \in S_{M_{C_x, I, A}} (\langle i, j \rangle \in r_{C_x, I, A}) \} \cup \\ \{ \langle C_x, I, A(i, v) \rangle : \exists M_{C_x, I, A} \in S_{M_{C_x, I, A}} (\langle i, v \rangle \in r_{C_x, I, A}) \},$$

gdzie $C_x, I, C(i)$ oznacza, że osobnik i jest wystąpieniem konceptu C w wystąpieniu kontekstu I dla kontekstu C_x , $C_x, I, R(i, j)$ oznacza, że para osobników (i, j) jest wystąpieniem roli R w wystąpieniu kontekstu I dla kontekstu C_x , oraz $C_x, I, A(i, v)$ oznacza, że wartość atrybutu A dla osobnika i w wystąpieniu kontekstu I dla kontekstu C_x jest równa v .

Aby określić opis świata dla ontologii kontekstowej z naszego przykładu, przyjmijmy, że odpowiedzi źródła danych dla poszczególnych zapytań zdefiniowanych w zbiorze odwzorowań S są następujące:

Dla zbioru odwzorowań konceptów:

- a) $r_{T_1, I_1, Kobieta} = \{17127903041\}$
b) $r_{T_1, I_1, Mezczyzna} = \{11057989675\}$

Dla zbioru odwzorowań ról:

- a) $r_{T_2, I_2, maStanowisko} = \{(11057989675, \text{Dyrektor})\}$
b) $r_{T_3, I_3, maDziecko} = \{(17127903041, 12030787123)\}$

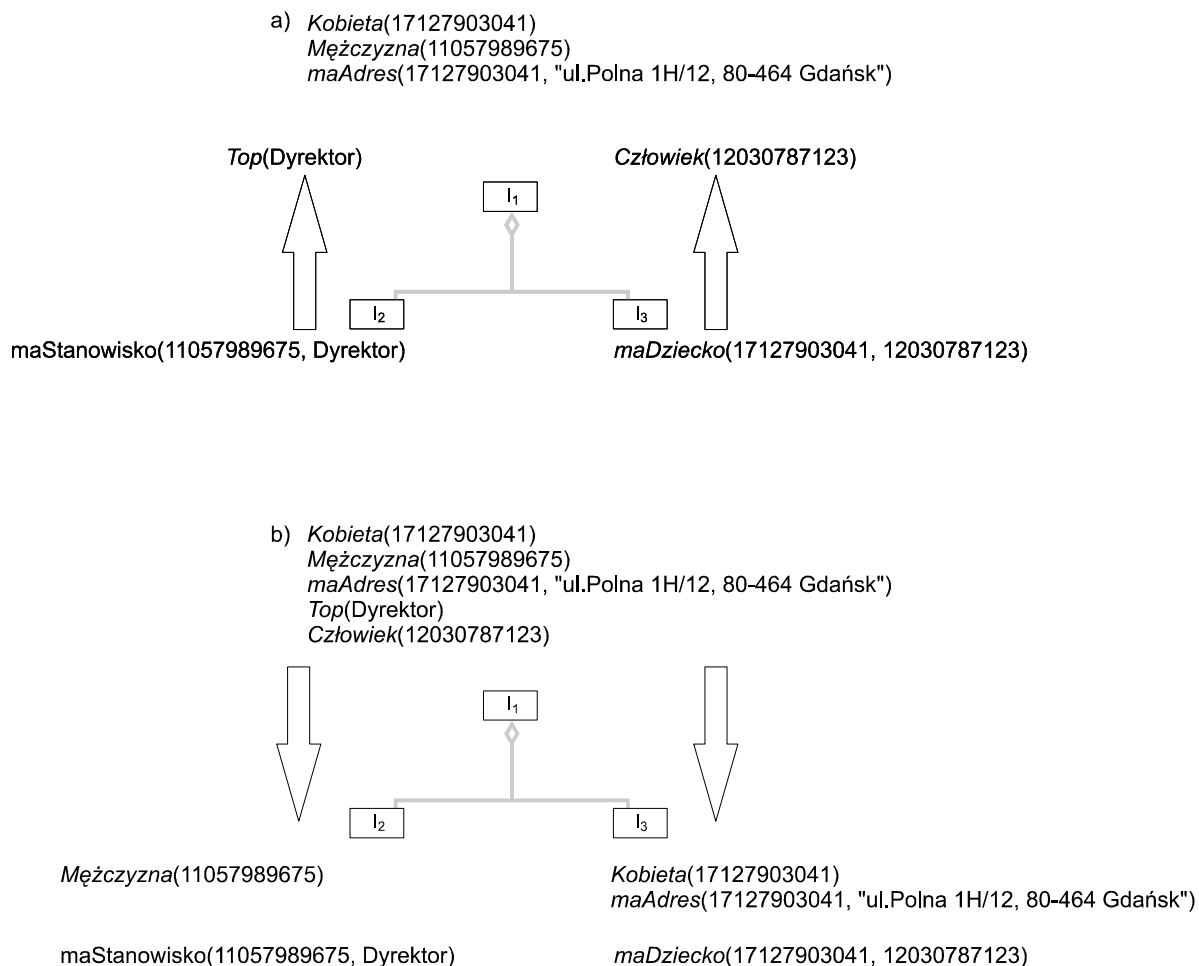
Dla zbioru odwzorowań atrybutów:

- a) $r_{T_1, I_1, maAdres} = \{(17127903041, \text{"ul.Polna 1H/12, 80-464 Gdańsk"})\}$

Przykładowa ontologia składa się z terminologii kontekstowej O_x zdefiniowanej na rysunku 9.6 oraz z opisu świata zdefiniowanego poniżej.

$T_1, I_1, Kobieta(17127903041)$,
 $T_1, I_1, Mezczyzna(11057989675)$
 $T_2, I_2, maStanowisko(11057989675, \text{Dyrektor})$,
 $T_3, I_3, maDziecko(17127903041, 12030787123)$,
 $T_1, I_1, maAdres(17127903041, \text{"ul.Polna 1H/12, 80-464 Gdańsk"})$

Na podstawie zdefiniowanego opisu świata pokażemy, jakie nowe fakty wywnioskuje baza wiedzy $\mathcal{K} = \langle O_x, O_s[S] \rangle$ z faktów pochodzących z innych kontekstów (dla jasności wyводу nie są opisywane nowe fakty wywnioskowane z aksjomatów już istniejących w danym kontekście). Proces przekazywania wniosków między kontekstami został przedstawiony na rysunku 9.7.



Rysunek 9.7. Przepływ faktów między kontekstami a) z dołu do góry i b) z góry w dół

Proces ten można podzielić na dwa etapy: przepływ wniosków z kontekstów podrzędnych do kontekstów nadrzędnych (z dołu do góry) i przepływ wniosków z kontekstów nadrzędnych do kontekstów podrzędnych (z góry do dołu). Proces z dołu do góry (rysunek 9.7a) polega na tym, że fakty zdefiniowane w terminach bardziej szczegółowych – terminach kontekstu podrzędnego – są przekształcane do terminów bardziej ogólnych – terminów kontekstu nadrzędnego. I tak baza wiedzy wywnioskuje w wystąpieniu kontekstu l_2 , że osobnik *Dyrektor* jest wystąpieniem konceptu *Stanowisko*. W kontekście T_1 jedyny koncept nadrzędny względem konceptu *Stanowisko* to koncept uniwersalny – *Top*. Dlatego do kontekstu l_2 dodawana jest asercja postaci *Top*(*Dyrektor*). Podobna sytuacja zachodzi w przypadku wystąpienia kontekstu l_3 . Na podstawie asercji *maDziecko*(17127903041, 12030787123) baza wiedzy wywnioskuje, że osobnik 12030787123 jest wystąpieniem konceptu *Człowiek*. Koncept ten jest zdefiniowany również w kontekście nadrzędnym T_1 , dlatego asercja *Człowiek*(12030787123) jest dodawana do wystąpienia kontekstu l_1 . Zgodnie z zasadą dziedziczenia w procesie z góry do dołu (rysunek 9.7b) z wystąpienia kontekstu nadrzędnego do wystąpienia kontekstu podrzędnego przechodzą wszystkie asercje należące

do dziedziny wystąpienia kontekstu podrzędnego. Dlatego do l_2 przechodzi asercja *Mężczyzna*(11057989675), a do l_3 dwie asercje *Kobieta*(17127903041) i *maAdres*(17127903041, "ul.Polna 1H/12, 80-464 Gdańsk").

Przedstawiony proces jest słuszny we wszystkich systemach wnioskujących stosujące kontekstualizację przestrzeni ontologicznej opartej na podejściu obiektowym. Dlatego też odwzorowania utworzone z wykorzystaniem metody SED muszą zostać przekształcone w taki sposób, aby wnioski przepływały poprawnie pomiędzy kontekstami. W tym celu rozszerzono metodę RED o proces wzbogacania odwzorowań kontekstowych. Proces wzbogacania odwzorowań kontekstowych polega na wzbogaceniu odwzorowań we wszystkich wystąpieniach kontekstów na podstawie odwzorowań z innych wystąpień kontekstów. Proces ten przebiega w następujących fazach:

- Wyznaczenie niejawnych odwzorowań konceptów na podstawie odwzorowań ról i atrybutów (analiza dziedziny i zakresu) niezależnie dla każdego wystąpienia kontekstu.
- Wyznaczenie odwzorowań dla konceptów uniwersalnych niezależnie dla każdego wystąpienia kontekstu.
- Wyznaczenie odwzorowań dla konceptów zdefiniowanych w kontekstach postaci $\exists R.C$ niezależnie dla każdego wystąpienia kontekstu.
- Wzbogacenie odwzorowań między kontekstami z dołu do góry.
- Wzbogacenie odwzorowań między kontekstami z góry do dołu.

Pierwsze trzy fazy procesu wzbogacania odwzorowań są niezależne od kontekstualizacji. Przebiegają analogicznie jak w przypadku ontologii niekontekstowej (traktując każde wystąpienie kontekstu określone w danym kontekście jak oddzielną ontologię). Sposób wzbogacania odwzorowań między kontekstami z dołu do góry zaprezentowano jako algorytm 9.1, zaś z góry do dołu jako algorytm 9.2.

Algorytm 9.1. Algorytm wzbogacania odwzorowań z dołu do góry

Wejście: Drzewo wystąpień kontekstów T ze zbiorem odwzorowań przypisanych do każdego węzła w drzewie.

Wyjście: Drzewo wystąpień kontekstów T ze zbiorem odwzorowań przypisanych do każdego węzła w drzewie wzbogaconych o odwzorowania wynikające z podrzędnych wystąpień kontekstów.

1. Stwórz listę l_1 i zapisz na tę listę wszystkie węzły będące liśćmi w drzewie T .
2. **Dopóki** lista l_1 nie jest pusta:
3. **Dla każdego** elementu z listy l_1 , który nie jest korzeniem drzewa T :
4. Stwórz pustą listę l_3 .
5. Pobierz węzeł nadrzędny p , kontekst Cx_p i wystąpienie kontekstu I_p .
6. **Dla każdego** odwzorowania konceptu C w aktualnie przetwarzanym elemencie z listy l_1 :
7. Pobierz zapytanie q_C .
8. **Jeżeli** w Cx_p jest zdefiniowany koncept C :
9. Dodaj koncept C do listy l_3 .
10. **W przeciwnym wypadku:**
11. Dodaj do listy l_3 wszystkie koncepty z Cx_p , po których koncept C bezpośrednio dziedziczy.
12. Zaktualizuj odwzorowania konceptów dla I_p :
- 12.a. **Dla każdego** konceptu D z listy l_3 :

- 12.b. **Jeżeli** w węźle p istnieje odwzorowanie
 $\mathcal{M}_D = \langle Cx_p, I_p, D, q_{Dp} \rangle$:
- 12.c. $q_{Dp} = q_{Dp}$ OR q_C .
- 12.d. **W przeciwnym wypadku:**
- 12.e. Stwórz odwzorowanie $\mathcal{M}_D = \langle Cx_p, I_p, D, q_C \rangle$.
13. Wyczyść listę l_3 .
14. **Koniec** {dla każdego}
15. **Dla każdego** odwzorowania roli R w aktualnie przetwarzanym elemencie z listy l_1 :
16. Pobierz zapytanie q_R .
17. **Jeżeli** w Cx_p jest zdefiniowana rola R :
18. Zaktualizuj odwzorowanie roli dla I_p :
- 18.a. **Jeżeli** w węźle p istnieje odwzorowanie
 $\mathcal{M}_R = \langle Cx_p, I_p, R, q_{Rp} \rangle$:
- 18.b. $q_{Rp} = q_{Rp}$ OR q_R .
- 18.c. **W przeciwnym wypadku:**
- 18.d. Stwórz odwzorowanie $\mathcal{M}_R = \langle Cx_p, I_p, R, q_R \rangle$.
19. **Koniec** {dla każdego}
20. **Dla każdego** odwzorowania atrybutu A w aktualnie przetwarzanym elemencie z listy l_1 .
21. Pobierz zapytanie q_A .
22. **Jeżeli** w Cx_p jest zdefiniowany atrybut A :
23. Zaktualizuj odwzorowanie atrybutu dla I_p :
- 24.a. **Jeżeli** w węźle p istnieje odwzorowanie
 $\mathcal{M}_A = \langle Cx_p, I_p, A, q_{Ap} \rangle$:
- 24.b. $q_{Ap} = q_{Ap}$ OR q_A .
- 24.c. **W przeciwnym wypadku:**
- 24.d. Stwórz odwzorowanie $\mathcal{M}_A = \langle Cx_p, I_p, A, q_A \rangle$.
25. **Koniec** {dla każdego}
26. **Jeżeli** węzła p nie ma na liście l_1 :
27. Dodaj węzeł p na listę l_1 .
28. Usuń aktualnie przetwarzany węzeł z listy l_1 .
29. **Koniec** {dla każdego}
30. **Koniec** {dopóki}

Algorytm 9.2. Algorytm wzbogacania odwzorowań z góry do dołu

Wejście: Drzewo wystąpień kontekstów T ze zbiorem odwzorowań przypisanych do każdego węzła w drzewie.

Wyjście: Drzewo wystąpień kontekstów T ze zbiorem odwzorowań przypisanych do każdego węzła w drzewie wzbogaconych o odwzorowania wynikające z nadrzędnych wystąpień kontekstów.

1. Stwórz listę l_1 i zapisz węzeł będący korzeniem w drzewie T na tę listę.
2. **Dopóki** lista l_1 nie jest pusta:
3. **Dla każdego** elementu z listy l_1 , który nie jest liściem drzewa T :
4. Pobierz listę l_3 węzłów podrzędnych
5. **Dla każdego** węzła p , kontekstu Cx_p i wystąpienia kontekstu I_p z listy l_3

6. **Dla każdego** odwzorowania konceptu C w aktualnie przetwarzanym elemencie z listy l_1 :
7. Pobierz zapytanie q_C .
8. Zaktualizuj odwzorowanie konceptu C dla I_p :
 - 8.a. **Jeżeli** w węźle p istnieje odwzorowanie

$$\mathcal{M}_D = \langle Cx_p, I_p, D, q_{Dp} \rangle:$$
 - 8.b. $q_{Dp} = q_{Dp}$ OR $(q_C \text{ INTERSECT } q_{TOPp})$.
 - 8.c. **W przeciwnym wypadku:**
 - 8.d. Stwórz odwzorowanie

$$\mathcal{M}_D = \langle Cx_p, I_p, D, q_C \text{ INTERSECT } q_{TOPp} \rangle.$$
9. **Koniec** {dla każdego}
10. **Dla każdego** odwzorowania roli R w aktualnie przetwarzanym elemencie z listy l_1 :
11. Pobierz zapytanie q_R .
12. Zaktualizuj odwzorowanie roli dla I_p :
 - 12.a. **Jeżeli** w węźle p istnieje odwzorowanie

$$\mathcal{M}_R = \langle Cx_p, I_p, R, q_{Rp} \rangle:$$
 - 12.b. $q_{Rp} = q_{Rp}$ OR $(q_R \text{ EXISTING IN } q_{TOPp})$.
 - 12.c. **W przeciwnym wypadku:**
 - 12.d. Stwórz odwzorowanie

$$\mathcal{M}_R = \langle Cx_p, I_p, R, q_R \text{ EXISTING IN } q_{TOPp} \rangle.$$
13. **Koniec** {dla każdego}
14. **Dla każdego** odwzorowania atrybutu A w aktualnie przetwarzanym elemencie z listy l_1 :
15. Pobierz zapytanie q_A .
16. Zaktualizuj odwzorowanie atrybutu dla I_p :
 - 16.a. **Jeżeli** w węźle p istnieje odwzorowanie

$$\mathcal{M}_A = \langle Cx_p, I_p, A, q_{Ap} \rangle:$$
 - 16.b. $q_{Ap} = q_{Ap}$ OR $(q_A \text{ EXISTING IN } q_{TOPp})$
 - 16.c. **W przeciwnym wypadku:**
 - 16.d. Stwórz odwzorowanie

$$\mathcal{M}_A = \langle Cx_p, I_p, A, (q_A \text{ EXISTING IN } q_{TOPp}) \rangle$$
17. **Koniec** {Dla każdego}
18. **Jeżeli** węzła p nie ma na liście l_1 :
19. Dodaj węzeł p na listę l_1 .
20. **Koniec** {dla każdego}
21. Usuń aktualnie przetwarzany węzeł z listy l_1 .
22. **Koniec** {dla każdego}
23. **Koniec** {dopółki}

W algorytmie 9.2 pojawia się formuła tworzenia nowego zapytania dla roli lub atrybutu w postaci $q \text{ EXISTING IN } q_{TOP}$. Formuła taka oznacza, że odpowiedzią na zapytanie $q \text{ EXISTING IN } q_{TOP}$ są tylko te pary zwracane w odpowiedzi na zapytanie q , w których wszystkie występujące nazwy osobników są zwracane w odpowiedzi na zapytanie q_{TOP} .

Wracając do omawianego przykładu, pokażemy, w jaki sposób odwzorowania podlegają wzbogaceniu. Po pierwszych 3 fazach zapytanie $q_{Człowiek}$ dla poszczególnych wystąpień kontekstów wygląda następująco:

$l_1: q_{Człowiek} = \text{select PESEL from LUDZIE where PELNOLETNI} = \text{'true'}$

$l_2: q_{Człowiek} = \text{select PESEL from LUDZIE, STANOWISKA where LUDZIE.IDSTANOWISKA} = \text{STANOWISKA.IDSTANOWISKA}$

$l_3: q_{Człowiek} = (\text{select R.PESEL from LUDZIE R, LUDZIE DZ, DZIECI where DZIECI.RODZIC} = \text{R.PESEL and DZIECI.DZIECKO} = \text{DZ.PESEL}) \text{ UNION } (\text{select DZ.PESEL from LUDZIE R, LUDZIE DZ, DZIECI where DZIECI.RODZIC} = \text{R.PESEL and DZIECI.DZIECKO} = \text{DZ.PESEL})$

Po 4. fazie zapytanie $q_{Człowiek}$ dla wystąpienia kontekstu l_1 ulega zmianie:

$l_1: q_{Człowiek} = (\text{select PESEL from LUDZIE where PELNOLETNI} = \text{'true'}) \text{ UNION } (\text{select PESEL from LUDZIE, STANOWISKA where LUDZIE.IDSTANOWISKA} = \text{STANOWISKA.IDSTANOWISKA}) \text{ UNION } (\text{select R.PESEL from LUDZIE R, LUDZIE DZ, DZIECI where DZIECI.RODZIC} = \text{R.PESEL and DZIECI.DZIECKO} = \text{DZ.PESEL}) \text{ UNION } (\text{select DZ.PESEL from LUDZIE R, LUDZIE DZ, DZIECI where DZIECI.RODZIC} = \text{R.PESEL and DZIECI.DZIECKO} = \text{DZ.PESEL})$

Nowo utworzone zapytanie algorytmem 9.1 jest sumą zapytań $q_{Człowiek}$ dla wszystkich kontekstów podrzędnych. Po 5. fazie, na skutek działania algorytmu 9.2, przykładowo w wystąpieniu kontekstu l_2 pojawi się odwzorowanie konceptu *Mężczyzna*:

$l_2: q_{Mężczyzna} = q_{Top} \text{ INTERSECT select PESEL from LUDZIE where PLEC} = \text{'m' and PELNOLETNI} = \text{'true'}, \text{ gdzie}$

$l_2: q_{Top} = (\text{select PESEL from LUDZIE, STANOWISKA where LUDZIE.IDSTANOWISKA} = \text{STANOWISKA.IDSTANOWISKA}) \text{ UNION } (\text{select NAZWA_STANOWISKA from LUDZIE, STANOWISKA where LUDZIE.IDSTANOWISKA} = \text{STANOWISKA.IDSTANOWISKA})$

Proces odpowiadania na zapytania jest analogiczny jak w przypadku ontologii niekontekstowej. Wynika to z faktu, że zapytanie jest zawsze zadawane do dokładnie jednego wystąpienia kontekstu, które jest traktowane wraz z kontekstem, którego jest wystąpieniem, jak niezależna ontologia, ze zdefiniowanym zbiorem odwzorowań.

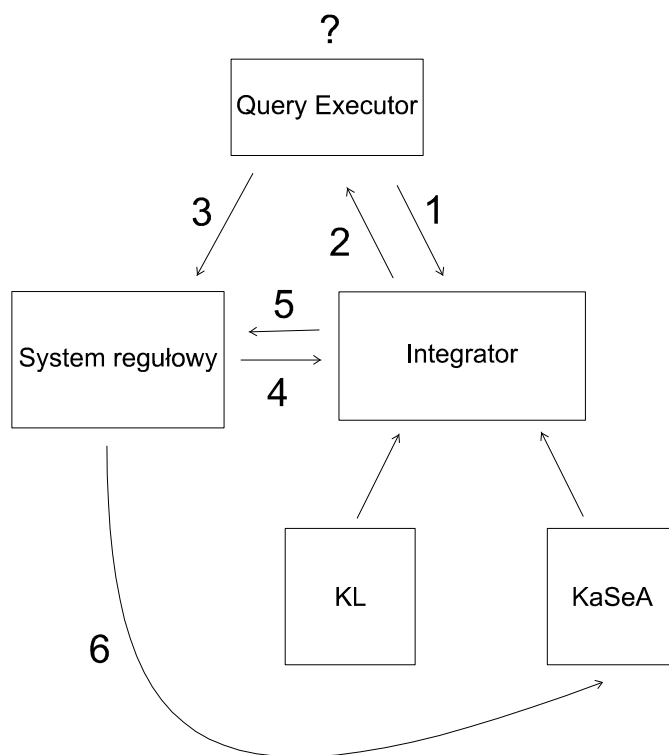
Budowanie źródeł wiedzy z wykorzystaniem metod ontologiczno-regułowych

Rozszerzenie funkcjonalności systemu KL o budowanie źródeł wiedzy z wykorzystaniem metod ontologiczno-regułowych wymaga zwiększenia ekspresywności wyrażania wiedzy o reguły. W tym celu planowano wykorzystanie języka SWRL. Proponuje się zwiększanie ekspresywności w dwóch krokach. W pierwszym kroku, tak jak i w aktualnej wersji systemu KL, przyjmuje się założenie UNA (ang. *Unique Name Assumption*). W języku SWRL każdy atom może być postaci: $C(x)$, $R(x, y)$, $A(x, v)$, $\text{sameAs}(x, y)$, $\text{differentFrom}(x, y)$ lub $\text{builtin}(r, x, \dots)$. Ze względu na przyjęcie w pierwszy kroku założenia UNA, zakładamy, że atomy nie mogą być postaci $\text{sameAs}(x, y)$ i $\text{differentFrom}(x, y)$. Dla tego kroku przedstawiono poniżej dwie wstępne propozycje wzbogacenia systemu KL o obsługę reguł. W kroku drugim proponowane jest odejście od założenia UNA. W tym celu zakładamy połączenie metod wzbogacania ekspresywności zaproponowanych dla pierwszego kroku z metodą UII.

Propozycja 1: Wykonywanie reguł w trakcie wykonywania zapytania

Możliwe jest wzbogacenie systemu KL o reguły w sposób pokazany na rysunku 9.8. Zapytanie jest zadawane zawsze do *Procesora zapytań*, który jest odpowiedzialny

za wykonanie zapytania. Ponadto zdefiniowano *System regułowy* odpowiedzialny za wnioskowanie z reguł, *Integrator* odpowiedzialny za integrowanie wiedzy z systemu KL i KaSeA, system KL odpowiedzialny za wnioskowanie ontologiczne z zewnętrznego źródła danych, system KaSeA odpowiedzialny za przechowywanie nowych faktów wywnioskowanych przez *System regułowy* i wnioskowanie z tych faktów. KaSeA przechowuje nowe wnioski dostarczone przez *System regułowy* i inicjalnie jej ABox jest pusty.



Rysunek 9.8. Wzbogacenie systemu KL o reguły

Zasadę działania systemu opartego na takiej architekturze przedstawimy na przykładzie zapytania o instance pewnego konceptu C . Zapytanie jest zadawane zawsze do *Procesora zapytań*. *Procesor zapytań* zadaje pytanie do *Integratora* (krok 1), w wyniku którego uzyskuje (krok 2) trzy zbiory osobników:

1. Zbiór $S_{\text{instances}}$ osobników, o których *Integrator* „wie”, że są wystąpieniami konceptu C ,
2. Zbiór $S_{\text{not_instances}}$ osobników, o których *Integrator* wie, że na pewno nie są one wystąpieniami konceptu C ,
3. Zbiór $S_{\text{maybe_instances}}$ osobników, co do których *Integrator* nie może jednoznacznie stwierdzić, czy są lub nie są wystąpieniami konceptu C .

Następnie dla wszystkich znanych osobników jest przeprowadzane wnioskowanie w oparciu o reguły (krok 3). Wnioskowanie to przebiega z udziałem *Integratora*, który odpowiada na zapytania ontologiczne (kroki 4, 5). W trakcie pracy *Systemu regułowego* wszystkie nowe wnioski są zapisywane do podsystemu KaSeA (krok 6). Wnioskowanie w *Systemie regułowym* jest powtarzane tak długo, aż do systemu KaSeA nie zostaną już dostarczone żadne nowe wnioski.

Można zauważyć, że wnioskowanie *Systemu regułowego* ma na celu podzielenie zbioru $S_{\text{maybe_instances}}$, zwróconego w kroku 1, na kolejne trzy podzbiory o analogicznym znaczeniu jak zbiory wymienione powyżej. Jedyna różnica polega na tym, że wiedza o osobnikach należących do tych konceptów jest już wiedzą z systemu ontologiczno-regułowego, nie zaś ontologicznego, jak w przypadku *Integratora*.

Niestety, takie podejście niesie za sobą poważne konsekwencje wydajności wykonywania zapytań. Przeprowadzenie wnioskowania w oparciu o reguły dla właściwie dowolnego osobnika może spowodować dodanie do systemu KaSeA nowego faktu, który z kolei może spowodować „lawinę” nowych wniosków ontologicznych lub regułowych, które mogą być niezbędne w procesie znajdowania odpowiedzi na interesujące nas zapytanie. Ograniczenie zbioru osobników o te osobniki, dla których nowe fakty ich dotyczące nie mają wpływu na znalezienie odpowiedzi na interesujące nas zapytanie, jest zadaniem bardzo trudnym. Dlatego zaproponowano inne podejście polegające na wykonywaniu reguł w procesie tworzenia odwzorowań dla systemu KL.

Propozycja 2: Wykonywanie reguł w trakcie tworzenia odwzorowań

W procesie tworzenia odwzorowań jest zapisywany zbiór odwzorowań pełnych i kompletnych. Niech $S_c = C_c \cup R_c \cup A_c$ będzie zbiorem konceptów, ról i atrybutów, dla których są zdefiniowane odwzorowania. Jeżeli dla konceptu, roli lub atrybutu jest zdefiniowane odwzorowanie, to jest to jednoznaczne z tym, że na pytanie o wystąpienie odwzorowanego konceptu, roli lub atrybutu system KL zwraca zawsze dokładnie dwa zbiory odpowiedzi: $S_{\text{instances}}$ i $S_{\text{not_instances}}$. Zbiór $S_{\text{maybe_instances}}$ jest zbiorem pustym. Przyjmijmy, że R_{eg} jest zbiorem reguł. Wiemy wtedy, że:

$$\begin{aligned} C \in C_c &\Rightarrow \neg \exists (r_{\text{eg}} \in R_{\text{eg}}) (r_{\text{eg}} = (\text{body} \Rightarrow C(x))), \\ R \in R_c &\Rightarrow \neg \exists (r_{\text{eg}} \in R_{\text{eg}}) (r_{\text{eg}} = (\text{body} \Rightarrow R(x,y))), \\ A \in A_c &\Rightarrow \neg \exists (r_{\text{eg}} \in R_{\text{eg}}) (r_{\text{eg}} = (\text{body} \Rightarrow A(x,v))). \end{aligned}$$

W procesie tworzenia odwzorowań wyznaczamy pełne i kompletne zapytania (lub procedurę ich uzyskania) dla konceptów, ról i atrybutów pojawiających się w zbiorze reguł R_{eg} po stronie *head*. Posłużmy się przykładem:

Przykład 9.1.

Mamy zewnętrzne źródło danych Z oraz terminologię T , którą jest opisane to źródło danych. Zakładamy, że zbiór odwzorowań S składa się z odwzorowań dla konceptów C , D i E :

$M_C = \langle C, q_C \rangle$ (konceptowi C przypisane jest zapytanie q_C do zewnętrznego źródła danych),

$M_D = \langle D, q_D \rangle$,

$M_E = \langle E, q_E \rangle$

odwzorowań dla ról R i P :

$M_R = \langle R, q_R \rangle$,

$M_P = \langle P, q_P \rangle$

oraz zbioru reguł R_{eg} :

(1) $F(x) \wedge R(x,y) \Rightarrow G(x)$,

(2) $G(x) \wedge C(x) \wedge P(x,y) \Rightarrow H(x)$.

System regułowy musi wyznaczyć odwzorowania:

$$M_G = \langle G, q_G \rangle$$

$$M_H = \langle H, q_H \rangle.$$

Wyznaczanie odwzorowań polega na wyszukaniu sposobu uzyskania zapytań q_F , q_G i q_H .

1. Dla zbioru $S = \{M_C, M_D, M_E, M_R, M_P\}$ przetwarzamy regułę (1):
 - a. $q_F = q_F(S_{\text{instances}})$, q_F zostało wyznaczone przez system KL na podstawie zbioru odwzorowań S (zbiór odwzorowań nie jest aktualizowany o M_F – zapytanie dla q_F jest zawsze wyszukiwane na żądanie na podstawie aktualnego zbioru S).
 - b. $q_G = q_G(S_{\text{instances}}) \vee (q_G(S_{\text{maybe_instances}}) \wedge q_F \wedge \text{first}(q_R))$, q_G zostało wyznaczone przez system KL na podstawie zbioru odwzorowań S i na podstawie reguły (2) (zbiór odwzorowań S jest aktualizowany o M_G , jakkolwiek M_G jest oznaczone jako niepełne i niekompletne).
2. Dla zbioru $S = \{M_C, M_D, M_E, M_R, M_P, M_G\}$ przetwarzamy regułę (2):
 - a. $q_H = q_H(S_{\text{instances}}) \vee (q_H(S_{\text{maybe_instances}}) \wedge q_G \wedge q_C \wedge \text{first}(q_P))$, q_H zostało wyznaczone przez system KL na podstawie zbioru odwzorowań S i na podstawie reguły (2) (zbiór odwzorowań S jest aktualizowany o M_H , jakkolwiek M_H jest oznaczone jako niepełne i niekompletne).

Procedura wyznaczania zapytań q_G i q_H jest powtarzana, dopóki żadne z tych zapytań nie ulegnie zmianie. Wszelkie zapytanie są zawsze wyznaczone na podstawie aktualnego zbioru odwzorowań S . W przypadku odwzorowań niepełnych i niekompletnych nie jest jednak zwracane bezpośrednio zapytanie zapisane w odwzorowaniu (zapytanie q_1), ale zapytanie będące iloczynem zapytania q_1 i zapytania utworzonego na podstawie zbioru odwzorowań pomniejszonego o odwzorowanie szukanego konceptu, roli lub atrybutu.

Budowanie odwzorowań polega na znalezieniu drogi utworzenia zapytania. Rozpatrzmy kolejny przykład:

Przykład 9.2.

Analogicznie jak w przykładzie 9.1, mamy zewnętrzne źródło danych Z oraz terminologię T , którą jest opisane to źródło danych. Zakładamy, że zbiór odwzorowań S składa się z dokładnie jednego odwzorowania dla roli *maRodzica*:

$$M_{\text{maRodzica}} = \langle \text{maRodzica}, q_{\text{maRodzica}} \rangle,$$

oraz zbioru reguł R_{eg} :

$$(1) \text{maRodzica}(x,y) \Rightarrow \text{maPrzodka}(x,y)$$

$$(2) \text{maPrzodka}(x,z) \Rightarrow \text{maRodzica}(x,y) \wedge \text{maPrzodka}(y,z)$$

System regułowy musi wyznaczyć odwzorowanie:

$$M_{\text{maPrzodka}} = \langle \text{maPrzodka}, q_{\text{maPrzodka}} \rangle$$

Zgodnie z krokami postępowania dla przykładu 9.1:

1. Dla zbioru $S = \{M_{\text{maRodzica}}\}$ dla reguły (1) $q_{\text{maPrzodka}} = q_{\text{maRodzica}}$
2. Dla zbioru $S = \{M_{\text{maRodzica}}, M_{\text{maPrzodka}}\}$ dla reguły (2)

$$q_{\text{maPrzodka}} = q_{\text{maRodzica}} \vee (\text{second}(q_{\text{maRodzica}}) = \text{second}(q_{\text{maPrzodka}}))$$

Zauważamy, że nie można zbudować zapytania dla $q_{\text{maPrzodka}}$, ponieważ zapytanie to zależy od samego siebie. W takich sytuacjach zapytania są tworzone w formie podobnej do reguły

uzyskania odpowiedzi na zapytanie.

9.3 Algorytmy przetwarzania zapytań w systemie integracji wiedzy

Z wymienionych w niniejszym rozdziale kierunków rozwoju przedstawionego w tej rozprawie systemu integracji wiedzy kierunkiem najmniej aktualnie zaawansowanym są algorytmy przetwarzania zapytań w ontologiczno-regułowym systemie integracji kontekstowych źródeł wiedzy. Zastosowanie reguł w systemie integracji wiedzy zostało zarysowane w [GRAB2004a]. Punkt ten został tutaj zamieszczony jako wskazanie drogi rozwoju w celu uzyskania kontekstowego systemu integracji wiedzy ontologiczno-regułowej, głównie jako zidentyfikowana potrzeba, a nie dojrzała idea rozwiązania zagadnienia.

9.4 Podsumowanie

Aktualnie rozwój opisanych tu metod jest na różnym poziomie zaawansowania. Te najbardziej dojrzałe zostały już zaimplementowane i wykorzystane w praktyce, część jest na etapie rozwoju koncepcyjnego, dla części zaś została dopiero zidentyfikowana potrzeba ich rozwoju.

Rozszerzenie metod SED i RED umożliwiające wnioskowanie z danych zewnętrznych zapisanych w różnych wystąpieniach kontekstów jest najbardziej dojrzałą z koncepcji przedstawionych w tym rozdziale. Koncepcja ta została zaimplementowana w systemie KISS opisanym w rozdziale 7 i sprawdzona dla kontekstowej ontologii leków w ramach projektu PIPS [PIPS2007]. Kontekstowe rozszerzenie metody SED i RED zostało również opublikowane w [GOCZ2008].

Pozostałe kierunki są rozwijane w ramach projektu „Metody zarządzania ontologiami i regułami w bazach wiedzy zgodnych z Semantic Web”, finansowanego przez MNiSzW – N N516 4115 33. Zarówno przedstawione kierunki rozwoju w zakresie modelu systemu integracji wiedzy, jak i metody budowania źródeł wiedzy są to dojrzejące koncepcje. Jedynie problemy dotyczące algorytmów przetwarzania zapytań są zidentyfikowanymi problemami, nad którymi prace nie zostały jeszcze podjęte.

Rozdział 10. Podsumowanie

W niniejszej rozprawie przedstawiono opracowane przez autora metody integracji wiedzy służące do konstrukcji systemu integracji wiedzy, który może być praktycznie stosowany we współczesnych systemach opartych na wiedzy. Do najważniejszych wyników rozprawy należą:

- zaproponowanie nowego modelu systemu integracji wiedzy,
- zdefiniowanie pojęcia źródła wiedzy i opracowanie sposobu jego tworzenia przez wzbogacenie warstwy danych o warstwę wiedzy,
- opracowanie algorytmów przetwarzania zapytań i ich walidacja w rzeczywistym systemie integracji wiedzy.

W rozprawie kolejne rozdziały służą osiągnięciu następujących celów rozprawy:

- 1) Przedstawienie modelu systemu integracji wiedzy.
- 2) Opracowanie algorytmów i metod służących do wytworzenia zamodelowanego systemu.
- 3) Wykazanie, że opracowane algorytmy i metody sprawdzają się w zastosowaniach praktycznych.

Osiągnięciu celu pierwszego rozprawy służy przede wszystkim rozdział trzeci, w którym przedstawiono model formalny wielowarstwowego systemu integracji wiedzy. Przedstawiony model systemu integracji wiedzy został oparty na modelu systemu integracji ontologii OIS zaproponowanym przez Calvanese. W odniesieniu do modelu OIS, w nowym modelu zdefiniowano:

- wielowarstwową architekturę wynikającą z zależności między terminologiami opisującymi wiedzę,
- sposób definiowania odwzorowań pomiędzy terminologiami,
- charakterystykę źródeł wiedzy integrowanych w ramach nowego systemu integracji,
- przyporządkowanie integrowanych źródeł wiedzy do odpowiedniej warstwy systemu,
- język zapytań do systemu integracji wiedzy.

Podstawową cechą zaproponowanego modelu systemu integracji wiedzy jest automatyzacja procesu definiowania struktury systemu. Struktura ta składa się z węzłów w hierarchii terminologii i przypisanych do tych węzłów źródeł wiedzy. Hierarchia terminologii jest budowana automatycznie na podstawie zależności między terminologiami, natomiast źródła wiedzy są automatycznie przypisywane do węzłów na podstawie terminologii, w terminach której opisana jest wiedza w danym źródle. W przedstawionym w pracy modelu przyjęto, że zależności pomiędzy terminologiami są opisane w postaci odwzorowań aksjomatycznych. Oznacza to, że zależność pomiędzy dwoma terminami z dwóch różnych terminologii może być wyrażona w postaci aksjomatu z pewnej logiki opisowej. Definiowanie struktury systemu integracji wiedzy może być zautomatyzowane również

wtedy, gdy przyjmimy inny sposób definiowania zależności między terminologiami, jak np. poprzez reguły.

Wielowarstwowa architektura systemu integracji wiedzy oparta na hierarchii terminologii ma zbiór cech umożliwiających jej stosowanie w rzeczywistych systemach opartych na wiedzy, co wynika z następujących przesłanek

- wiedza może być przechowywana w różnych źródłach wiedzy opisanych różnymi terminologiami,
- zapytania mogą być kierowane do wybranego węzła zarządzającego terminami użytymi w zapytaniu.

Szczególną rolę odgrywa pierwsza z wymienionych cech pozwalająca każdemu użytkownikowi na tworzenie źródeł wiedzy i podłączanie ich do systemu integracji wiedzy. To założenie w sposób szczególny wpasowuje się w ideę Sieci Semantycznej, w której każdy użytkownik powinien mieć możliwość udostępniania wiedzy w sposób analogiczny, jak ma możliwość umieszczania stron WWW w Internecie. Każdy użytkownik może utworzyć swoje własne źródła wiedzy. Źródła te zostaną automatycznie skojarzone z odpowiednim węzłem w hierarchii terminologii. Jedyny warunek, jaki takie źródła muszą spełniać, to implementacja zdefiniowanego w modelu języka zapytań. Druga wymieniona cecha ma szczególny wpływ na wydajność przetwarzania zapytań. W odróżnieniu od modelu Calvanese, w którym zapytania są zadawane do ontologii globalnej, w modelu zaproponowanym w rozprawie zapytanie może być zadane do dowolnego węzła w hierarchii oraz do dowolnych źródeł wiedzy.

Bardzo istotne znaczenie ma również elastyczność zaproponowanego modelu. Jak już wspomniano, możliwe jest automatyczne definiowanie struktury systemu niezależnie od sposobu definiowania odwzorowań pomiędzy terminologiami. Kwestia ta jest istotna, ponieważ wraz z rozwojem systemów zarządzania wiedzą stawiane są również nowe wymagania. Standardowo zależności między terminologiami budowane są w postaci aksjomatów logiki opisowej wykorzystując relację importu. Jednak coraz częściej widać potrzebę wykorzystywania również ekspresywności reguł do budowania odwzorowań pomiędzy ontologiami. Elastyczność zaproponowanego modelu nie ogranicza się jednak tylko do sposobu definiowania odwzorowań między terminologiami. Wyraźnie widać, że w Sieci Semantycznej nie stworzono jeszcze języka zapytań dla systemów zarządzania wiedzą spełniającego większość wymagań. Standardem W3C jest język SPARQL [SPARQL2008], jest on jednak językiem przeznaczonym do operowania na trójkach RDF, nie zaś na ontologii wyrażonej w logice opisowej. W modelu przedstawionym w rozprawie zaproponowano wykorzystanie języka DIGUT opartego na języku DIG 1.1. Język ten dostarcza podstawowych metod dostępu do wiedzy zapisanej w logice opisowej (zarówno w zakresie zapytań terminologicznych, jak i asercyjnych), jednak nie jest pozbawiony zasadniczych wad. Trudno w nim wyrazić zapytania złożone, a definiowanie zapytań zagnieżdżonych nie jest w ogóle możliwe. Prowadzone są prace nad innymi językami takimi jak.: OWL-QL [OWLQL2003], [OWLQLSO], *n*RQL [RACER], czy DIG 2.0 [DIG2.0], [DIG2.0QI]. Żaden z tych języków nie został jeszcze zaakceptowany jako obowiązujący standard. Dlatego tak ważna jest elastyczność modelu systemu integracji wiedzy.

Możliwość przededefiniowania w istniejącym modelu pewnych jego elementów nie stoi jednak w sprzeczności do jego szerokiej praktycznej użyteczności w aktualnej jego wersji, co pokazano przez zastosowanie tego modelu w systemie PIPS, dla celów związanych z zarządzaniem wiedzą dotyczącą odżywiania. Jednak oczywiste jest, że przedstawiony w niniejszej rozprawie model będzie ewoluował w różnych kierunkach zdefiniowanych w rozdziale 9, z których do najważniejszych należy zwiększenie ekspresywności ontologii o reguły.

Rozdziały czwarty, piąty i szósty służą wypełnieniu celu drugiego rozprawy. Rozdział czwarty wprowadza Czytelnika w kartograficzną reprezentację wiedzy, która jest podstawą metod i algorytmów przedstawionych w rozdziałach piątym i szóstym, opisujących opracowanie zamodelowanego systemu.

W niniejszej rozprawie kartograficzna reprezentacja wiedzy została przyjęta jako podstawa metod i algorytmów, ze względu na to, że :

- dostarcza metod efektywnego przechowywania wiedzy w relacyjnej bazie danych,
- dostarcza efektywne algorytmy realizacji zapytań do bazy wiedzy,
- jest zweryfikowana w praktycznych zastosowaniach,
- jest intensywnie rozwijana w zakresie ekspresywności obsługiwanych ontologii (również w kierunku obsługi reguł).

System integrowania wiedzy jest systemem, który w swoim założeniu kieruje zapytania do różnych źródeł wiedzy i łączy uzyskane odpowiedzi częściowe w jedną spójną odpowiedź. Źródło wiedzy jest zdefiniowane jako system zarządzania wiedzą, implementujący zdefiniowany w modelu język zapytań. Dlatego, z punktu widzenia systemu integracji wiedzy, kluczowa jest wydajność algorytmów wnioskujących zaimplementowanych zarówno w źródłach wiedzy, jak i w podsystemach łączenia odpowiedzi uzyskanych z różnych źródeł.

Metoda kartograficzna sprawdziła się w systemie PIPS zarządzania wiedzą medyczną. Również przeprowadzone testy wydajności algorytmów wnioskujących z wiedzy reprezentowanej kartograficznie wskazują na ich praktyczną użyteczność i lepszą skalowalność względem liczby osobników niż systemy konkurencyjne, takie jak RacerPro [HAAR2001] czy Pellet [SIRI2004]. Uzyskane wyniki dawały uzasadnioną nadzieję, że zastosowanie metody kartograficznej do zbudowania źródeł wiedzy dla zewnętrznych źródeł danych i w algorytmach przetwarzania zapytań również przyniesie podobne wyniki.

W rozdziale piątym przedstawione zostały algorytmy i metody umożliwiające budowanie źródeł wiedzy dla zewnętrznych źródeł danych. Jest to kluczowe zagadnienie dla celów tej rozprawy, ponieważ wiedza może być zapisana w różnych źródłach. Ujednolicenie dostępu do tych źródeł w systemie integracji wiedzy zostało osiągnięte za pomocą metod SED (*Semantic Enrichment of Data*) i RED (*Reasoning over External Data*).

Celem metody SED jest przekształcenie zewnętrznego źródła danych w ontologię. Przekształcenie to nie jest faktycznie wykonywane, a jedynie jest tworzone na poziomie abstrakcyjnym. W metodzie SED w pierwszej kolejności jest wybierana terminologia, a następnie dane zapisane w źródle danych są opisywane semantycznie w taki sposób, aby mogły być traktowane jak zbiory asercji. Opis semantyczny danych jest tworzony w postaci par: (termin z terminologii, zapytanie do zewnętrznego źródła danych), nazwanych odwzorowaniami. Takie podejście umożliwia stosowanie metody SED względem dowolnego źródła danych implementującego język zapytań. Niezależność od typu źródła danych jest podstawową cechą metody SED. Aktualnie większość danych nie jest przechowywana w źródłach wiedzy, ale właśnie w źródłach danych, z których najbardziej popularne są serwery relacyjnych baz danych i bazy danych w formacie XML. Istotnym osiągnięciem rozprawy jest opracowanie metody, która pozwala na jednolite postrzeganie przez system integrujący dowolnego źródła danych tego typu. Jest to niezwykle ważne zagadnienie również w odniesieniu do inicjatywy Sieci Semantycznej. Aby Internet mógł stać się Siecią Semantyczną, muszą istnieć mechanizmy przekształcające dane zapisane w Internecie w wiedzę. Jednym z takich mechanizmów jest właśnie metoda SED. Dodatkowo, połączenie prac wykonanych w ramach Sieci Semantycznej dotyczących semantycznego adnotowania stron HTML z metodą SED umożliwi również przekształcanie danych zawartych na stronach HTML w wiedzę. Z niezależności metody SED od typu źródła danych wynika, że również repozytoria RDF mogą być traktowane jako takie źródła (językiem dostępu do repozytoriów

RDF jest język SPARQL). Wykorzystując mechanizmy RDF (*Resource Description Framework attributes*) [RDFaP], [RDFaSP], możliwe jest adnotowanie stron HTML w taki sposób, aby dane w nich zawarte mogły być postrzegane jako zbiór trójek RDF. Użycie metody SED pozwala zatem na tworzenie na poziomie koncepcyjnym ontologii z właściwie dowolnego źródła danych.

Celem metody RED jest dostarczenie algorytmów wnioskujących z abstrakcyjnie utworzonej ontologii na podstawie odwzorowań zdefiniowanych przy użyciu metody SED. W metodzie RED wykorzystano kartograficzną reprezentację wiedzy w celu osiągnięcia dobrej efektywności opracowanych algorytmów oraz skalowalności względem liczby asercji (przekładających się na wielkość danych zapisanych w zewnętrznych źródłach danych). Pomimo nie najlepszej złożoności obliczeniowej zaproponowanych algorytmów, cele te zostały osiągnięte. Wynika to przede wszystkim ze specyfiki ontologii i zastosowanych optymalizacji. Przypadki zapytań, w których algorytmy wnioskujące osiągają pesymistyczną złożoność obliczeniową, są bardzo rzadkie, zaś zastosowane optymalizacje polegają na zapamiętywaniu treści zapytań złożonych, dzięki czemu odpowiedź na zapytanie sformułowane po raz kolejny jest zależna tylko od szybkości odpowiedzi zewnętrznych źródeł danych.

Rozdział szósty opisuje algorytmy przetwarzania zapytań w systemie integracji wiedzy. Kluczowym algorytmem przetwarzania zapytań zaproponowanym w rozprawie jest algorytm SPIDER. W algorytmach wnioskujących z wiedzy reprezentowanej kartograficznie najważniejsze jest wyznaczenie sygnatury osobnika lub grupy osobników. Również celem algorytmu SPIDER jest wyznaczenie sygnatury osobnika. W algorytmie SPIDER sygnatura ta musi być wyznaczona na podstawie wiedzy o osobniku zapisanej w różnych źródłach wiedzy. Najważniejszym zadaniem tego algorytmu jest wyznaczenie sygnatury złożonej, która przenosi nie tylko wiedzę zebraną ze źródeł wiedzy, ale również wiedzę wywnioskowaną z wiedzy zapisanej w źródłach wiedzy. Cel ten został osiągnięty w takim zakresie, w jakim pozwala na to metoda kartograficzna. Systemy wnioskujące oparte na kartograficznej reprezentacji wiedzy nie gwarantują kompletności odpowiedzi na zapytania asercjonalne. Nie udzieleni przez te systemy odpowiedzi kompletnej zachodzi jednak niezwykle rzadko i nie przeczy użyteczności tych algorytmów.

Przedstawienie metod integracji wiedzy nie jest jednak wystarczające do udowodnienia tezy rozprawy. Musi zostać spełniony również trzeci cel rozprawy. Aby ten cel został osiągnięty, po pierwsze przedstawiono system wykorzystujący metody integracji wiedzy zaproponowane w rozprawie, a po drugie pokazano zastosowanie tego systemu w praktyce. System ten, nazwany KISS, został opisany w rozdziale siódmym. W systemie KISS źródłami wiedzy są systemy: KaSeA i Knowledge Layer. System KaSeA jest systemem wnioskowania opartym na kartograficznej reprezentacji wiedzy. System Knowledge Layer jest systemem wnioskowania z zewnętrznych źródeł danych implementującym metody SED i RED. Podsystemem łączenia odpowiedzi z tych źródeł wiedzy jest External Query Manager. Implementacja systemu KISS została wykonana dla celów systemu zarządzania wiedzą medyczną. Rozdział ósmy pokazuje praktyczne wykorzystanie systemu KISS do integracji wiedzy w usłudze doboru i wspomagania stosowania optymalnej diety w systemie PIPS, zrealizowanym w projekcie zintegrowanym 6. Programu Ramowego Unii Europejskiej PIPS. Wykorzystanie systemu KISS pokazano na rzeczywistych ontologiach i źródłach wiedzy.

Na potrzeby omawianej usługi w systemie PIPS została utworzona hierarchia sześciu terminologii. Do węzłów w hierarchii terminologii zostało podłączonych pięć źródeł wiedzy, z których cztery były relacyjnymi źródłami danych wzbogaconymi o warstwę wiedzy – Knowledge Layer, a jedno źródło wiedzy było źródłem ontologicznym, z którego wiedza zapisana była w bazie wiedzy systemu KaSeA. Konfiguracja systemu KISS dla systemu PIPS obejmowała źródła danych o różnej wielkości (z których niektóre zawierały bardzo dużą

liczbę krotek). Taka konfiguracja pozwoliła sprawdzić użyteczność zaproponowanych w rozprawie metod i algorytmów przy różnorodnej charakterystyce zewnętrznych źródeł danych. Również zestaw tzw. zapytań kompetencyjnych (czyli zapytań występujących w możliwych scenariuszach użycia systemu), dla których przeprowadzono testy mierzące średnie czasy odpowiedzi, obejmował różne typy zapytań asercjonalnych i terminologicznych. System KISS był używany przez agenty na potrzeby portalu PIPS. Osiągnięte wyniki pokazały, że zaproponowane metody sprawdzają się w poważnych przedsięwzięciach informatycznych.

Tym samym, w rozprawie pokazano, w jaki sposób zostały osiągnięte wszystkie trzy cele rozprawy, tym samym teza rozprawy:

Zaproponowane w rozprawie metody łączenia ontologii i algorytmy przetwarzania zapytań w rozproszonych bazach wiedzy pozwalają na konstrukcję systemu integracji wiedzy o parametrach jakościowych i ilościowych umożliwiającym jego praktycznie zastosowanie we współczesnych systemach opartych na wiedzy

została dowiedziona.

Rozprawa kończy się rozdziałem opisującym dalsze kierunki prac nad systemem integracji wiedzy. Na polu rozwoju modelu systemu integracji wiedzy jest to przede wszystkim rozwój systemu w kierunku integrowania kontekstowych źródeł wiedzy, jako że inżynierowie ontologii dążą do ich modularyzacji. Równie ważnym aspektem jest rozszerzenie systemu o obsługę reguł, zarówno w zakresie ekspresywności ontologii, jak i wyrażania zależności między terminami z różnych terminologii. W niniejszej rozprawie nie został też rozwiązany problem określania tożsamości osobników, gdyż wszystkie metody zaproponowane w rozprawie bazują na założeniu unikatowości nazw (UNA). Oddzielnym zagadnieniem jest połączenie prac związanych z integrowaniem źródeł wiedzy z zagadnieniami dotyczącymi wiarygodności wiedzy w systemach zarządzania wiedzą.

Prace nad systemem integracji i rozwojem zaproponowanych metod toczą się dalej w ramach grantu badawczego MNiSzW „Metody zarządzania ontologiami i regułami w bazach wiedzy zgodnych z Semantic Web” nr N N516 4115 33.

Słownik pojęć

ABox	Opis świata, w którym definiujemy osobniki oraz asercje binarne i unarne o tych osobnikach.
aksjomat	Zdanie definiujące zależności logiczne między konceptami, rolami i atrybutami.
aksjomat równości	Zdanie definiujące równość dwóch konceptów, ról lub atrybutów.
aksjomat terminologiczny	Patrz aksjomat.
aksjomat zawierania	Zdanie definiujące relację zawierania między dwoma konceptami lub rolami.
asercja	Zdanie opisujące cechę osobnika.
asercja wystąpienia konceptu	Zdanie stwierdzające przynależność osobnika do pewnego konceptu.
asercja wystąpienia roli	Zdanie stwierdzające powiązanie rolą dwóch osobników.
dopełnienie roli	Osobnik, z którym inny osobnik pozostaje w relacji.
dziedzina interpretacji	Zbiór nazw osobników.
dziedzina konkretna	Para $\mathcal{D} = (\Delta^{\mathcal{D}}, \text{pred}(\mathcal{D}))$, gdzie zbiór $\Delta^{\mathcal{D}}$ reprezentuje dziedzinę, a zbiór $\text{pred}(\mathcal{D})$ składa się z nazw predykatów operujących na tej dziedzinie. Z każdą nazwą predykatu $P \in \text{pred}(\mathcal{D})$ jest skojarzona krotność n oraz dla każdego n -arnego predykatu $P^{\mathcal{D}}$ zachodzi relacja $P^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$.
funkcja interpretacji	Funkcja, która każdemu konceptowi A przypisuje zbiór $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, a każdej roli atomowej R binarną relację $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.
funkcja przyporządkowująca	Funkcja przyporządkowująca każdemu węzłowi v_i terminologię \mathcal{O}_{x_i} ze zbioru terminologii $S_{\mathcal{O}}$: $f(v_i) = \mathcal{O}_{x_i}$.
GAV	<i>Global As View</i> : sposób budowania odwzorowań pomiędzy schematem globalnym i lokalnym, w którym terminy ze schematu globalnego są wyrażone w postaci zapytań do schematu lokalnego.

hierarchia terminologii	<p>Hierarchią terminologii \mathcal{H}_T nazywamy graf skierowany, w którym:</p> <ol style="list-style-type: none"> 1) istnieje m węzłów, gdzie $m = S_O$ jest mocą zbioru terminologii S_O, 2) różnowartościowa funkcja przyporządkowująca f przyporządkowuje każdemu węzłowi v_i terminologię O_{x_i} ze zbioru terminologii S_O: $f(v_i) = O_{x_i}$, <p>przy czym krawędź e_{ij} skierowana od wierzchołka v_i do v_j oznacza, że terminologia $O_{x_i} = f(v_i)$ importuje terminologię $O_{x_j} = f(v_j)$.</p>
hierarchia źródeł wiedzy	Przypisanie źródeł wiedzy do węzłów w hierarchii terminologii.
interpretacja	Interpretacja \mathcal{I} jest parą: dziedzina interpretacji i funkcja interpretacji.
język opisu	Zbiór konstruktorów w logice opisowej.
KaSeA	<i>Knowledge Siganture Analyser</i> , system zarządzania wiedzą oparty na kartograficznej reprezentacji wiedzy.
KISS	<i>Knowledge Integration Support System</i> , system integracji wiedzy.
KL	<i>Knowledge Layer</i> , system zarządzania wiedzą zapisaną w zewnętrznych źródłach danych.
kompletność	Odpowiedź na zapytanie jest kompletna, jeżeli zapytanie zwaraca wszystkie osobniki spełniające warunki zadane w zapytaniu.
koncept atomowy	Koncept pierwotny, nazwany.
koncept pusty	Pusty zbiór osobników.
koncept uniwersalny	Zbiór wszystkich osobników z dziedziny zainteresowania.
konstruktor	Sposób zapisania relacji pomiędzy konceptami lub rolami.
kontekst	Punkt widzenia na daną dziedzinę wiedzy.
kontekstowe źródło wiedzy	Źródło wiedzy opisane ontologią kontekstową.
kwantyfikacja ogólna	Koncept postaci $\forall R.C$.
kwantyfikacja szczegółowa	Koncept postaci $\exists R.C$.

LAV	<i>Local As View</i> : sposób budowania odwzorowań pomiędzy schematem globalnym i lokalnym, w którym terminy ze schematu lokalnego są wyrażone w postaci zapytań do schematu globalnego.
łączenie ontologii	Proces znajdowania wspólnych terminów pomiędzy różnymi terminologiami i wywiedzenie z nich nowej terminologii wynikowej.
MCA	<i>Maximum Coverage Algorithm</i> , algorytm wyznaczania wyrażenia sygnaturowego składającego się z sygnatur odwzorowywanych dla zadanej sygnatury.
ModA	Moduł odpowiedzialny za wystąpienie kontekstu w kontekstowym systemie integracji wiedzy.
ModT	Moduł odpowiedzialny za kontekst w kontekstowym systemie integracji wiedzy.
mapa konceptów	Mapą konceptów dla dialektu $C\mathcal{LC}(T)$ nazywamy funkcję $s: C(C\mathcal{LC}(T)) \rightarrow \mathbb{B}^n$ przypisującą każdemu konceptowi wyrażonemu w $C\mathcal{LC}(T)$ sygnaturę o długości n i mającą następującą właściwość: $(C, D \in C(C\mathcal{LC}(T))): s(C) \leq s(D) \Leftrightarrow T \models C \sqsubseteq D$.
model aksjomatu	Interpretacja spełniająca aksjomat.
model asercji	Interpretacja spełniająca asercję.
model ontologii	Interpretacja spełniająca wszystkie aksjomaty i asercje zdefiniowane w ontologii.
negacja konceptu	Koncept postaci $\neg C$, gdzie C jest konceptem.
odwzorowanie aksjomatyczne	Aksjomat definiujący zależności pomiędzy terminami z dwóch różnych terminologii.
odwzorowanie atrybutu	Odwzorowaniem atrybutu $\mathcal{M}_A = \langle A, q_A \rangle$ nazywamy parę składającą się z atrybutu A wyrażonego w logice opisowej \mathcal{L}_O na alfabecie \mathcal{A}_{O_A} i zapytania q_A do źródła danych \mathcal{Z} , dla którego odpowiedzią $r_A = \{ \langle i, v \rangle : i \text{ jest nazwą osobnika, a } v \text{ pewną wartością} \}$ jest zbiór par: nazwa osobnika, wartość.

odwzorowanie konceptu	<p>Odwzorowaniem konceptu $\mathcal{M}_C = \langle C, q_C \rangle$ nazywamy parę składającą się z konceptu C, atomowego lub złożonego, wyrażonego w logice opisowej \mathcal{L}_O na alfabecie $\mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A}$ i zapytania q_C do źródła danych \mathcal{Z}, dla którego odpowiedzią $r_C = \{i: i \text{ jest nazwą osobnika}\}$ jest zbiór nazw osobników.</p>
odwzorowanie roli	<p>Odwzorowaniem własności $\mathcal{M}_R = \langle R, q_R \rangle$ nazywamy parę, składającą się z roli R, atomowej lub złożonej wyrażonej w logice opisowej \mathcal{L}_O na alfabecie \mathcal{A}_{O_R} i zapytania q_R do źródła danych \mathcal{Z}, dla którego odpowiedzią $r_R = \{\langle i, j \rangle: i, j \text{ są nazwami osobników}\}$ jest zbiór par nazw osobników.</p>
ograniczenie liczebnościowe	<p>Koncept postaci $\geq n R$ lub $\leq n R$.</p>
ontologia	<p>Ontologią O wyrażoną w pewnej logice opisowej \mathcal{L}_O nazywamy parę $\langle O_x, O_s \rangle$. Niech \mathcal{A}_O oznacza alfabet terminów zdefiniowanych w ontologii O. $\mathcal{A}_O = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A} \cup \mathcal{A}_{O_I}$ jest zbiorem terminów zdefiniowanych w czterech alfabetach: alfabecie \mathcal{A}_{O_C} terminów będących nazwami konceptów, alfabecie \mathcal{A}_{O_R} terminów będących nazwami ról, alfabecie \mathcal{A}_{O_A} terminów będących nazwami atrybutów i alfabecie \mathcal{A}_{O_I} terminów będących nazwami osobników. Wówczas O_x (TBox, terminologia ontologii O) jest zbiorem aksjomatów wyrażonych w terminach alfabetu $\mathcal{A}_{O_T} = \mathcal{A}_{O_C} \cup \mathcal{A}_{O_R} \cup \mathcal{A}_{O_A}$, zaś O_s (ABox, asercje dotyczące osobników ontologii O) jest zbiorem asercji wyrażonych w terminach alfabetu \mathcal{A}_O.</p>
operator epistemiczny	<p>Operator modalny oznaczający, że „baza wiedzy wie, że...”.</p>
opis świata	<p>Patrz ABox.</p>

pajęczyna	<p>Pajęczyną $P: \mathcal{O}_{x_n} \rightarrow \mathcal{O}_{x_p}$ pomiędzy terminologią nadrzędną \mathcal{O}_{x_n} a terminologią podrzędną \mathcal{O}_{x_p} nazywamy przyporządkowanie $p(o_n) = o_p$ przyporządkowujące obszarom z mapy konceptów terminologii \mathcal{O}_{x_n} obszary z mapy konceptów terminologii podrzędnej \mathcal{O}_{x_p}.</p> <p>Przyporządkowanie p musi spełniać następujący warunek:</p> $\forall (I = MD(\mathcal{O}_{x_n}) \wedge I = MD(\mathcal{O}_{x_p})) (C(o_p)^I \subseteq C(o_n)^I),$ <p>gdzie $C(o)$ oznacza dowolny koncept odpowiadający obszarowi o.</p>
podmiot roli	Osobnik powiązany z innym osobnikiem pewną relacją.
poprawność	Odpowiedź na zapytanie jest poprawna, jeżeli wszystkie osobniki zwrócone w odpowiedzi spełniają warunki określone w zapytaniu.
problem określenia zbioru wystąpień konceptu	Jakie osobniki należą do konceptu C ?
problem rozłączności	Czy zbiory konceptów C i D są zawsze rozłączne? (Czy $C \sqcap D \equiv \perp$?)
problem równoważności	Czy zbiory wystąpień konceptów C i D są zawsze równe? (Czy $C \equiv D$?)
problem spełnialności	Czy koncept C może mieć wystąpienia? (Czy $C \equiv \perp$?)
problem sprawdzenia przynależności	Czy dany osobnik należy do konceptu C ?
problem sprawdzenia spójności	Czy baza wiedzy ma niepusty model?
problem zawierania	Czy zbiór wystąpień konceptu C jest zawsze podzbiorem wystąpień konceptu D ? (Czy $C \sqsubseteq D$?)
RED	<i>Reasoning over External Data</i> : metoda wnioskowania z zewnętrznych źródeł danych.
SED	<i>Semantic Enrichment of Data</i> : metoda semantycznego opisu danych zapisanych w zewnętrznych źródłach danych.
Sieć Semantyczna	Sieć WWW wzbogacona o elementy semantyczne rozpoznawane przez komputery.
SPIDER	Algorytm przetwarzania zapytań do zewnętrznych źródeł wiedzy.

suma konceptów	Koncept postaci $C \sqcup D$.
sygnatura	Sygnaturą o długości n , $n \in \mathbb{N}$, nazywamy element zbioru $\mathbb{B}^n = \{0,1\}^n$. Sygnatura s o długości n jest ciągiem n -elementowym.
sygnatura nadrzędna	Sygnaturą nadrzędną s_n osobnika o dla pajęczyny $P(O_{x_n}, O_{x_p})$ nazywamy sygnaturę osobnika wyznaczoną dla terminologii nadrzędnej O_{x_n} i odpowiadającą sygnaturze osobnika o dla terminologii O_{x_p} dla źródeł wiedzy P_{KS} .
sygnatura podrzędna	Sygnaturą podrzędną s_p osobnika o dla pajęczyny $P(O_{x_n}, O_{x_p})$ nazywamy sygnaturę osobnika wyznaczoną dla terminologii podrzędnej O_{x_p} dla źródeł wiedzy P_{KS} i równą sygnaturze złożonej $s_c(o, f^l(O_{x_p})[\mathcal{H}_T, P_{KS}])$ osobnika o w węźle $f^l(O_{x_p})$ w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS} .
sygnatura prosta	Sygnaturą prostą osobnika o w węźle v w hierarchii terminologii \mathcal{H}_T dla źródeł wiedzy ze zbioru P_{KS} nazywamy sygnaturę s_s równą przecięciu sygnatur osobnika o wyznaczonych dla poszczególnych źródeł wiedzy $KS \in P_{KS}$ i przypisanych do węzła v .
sygnatura złożona	Niech $f(v) = O_{x_v}$, a $S_P = \{P: P(O_{x_v}, O_{x_p})\}$ jest zbiorem pajęczyn, w których O_{x_v} jest terminologią nadrzędną. Dodatkowo $S_{O_{x_p}} = \{O_x: P(O_{x_v}, O_x) \in S_P\}$. Sygnaturą złożoną $s_c(o, v[\mathcal{H}_T, P_{KS}])$ nazywamy przecięcie sygnatury prostej $s_s(o, v[\mathcal{H}_T, P_{KS}])$ z sygnaturami podrzędnymi $s_p[o, P(O_{x_n}, O_{x_p}), P_{KS}]$, gdzie $O_{x_p} \in S_{O_{x_p}}$.
TBox	Terminologia, zbiór aksjomatów.
terminologia	Patrz TBox.
unikatowość nazw	<i>Unique Name Assumption</i> – UNA, dwa osobniki identyczne semantycznie mają tę samą nazwę.
UII	<i>Unique Individual Identification</i> : metoda identyfikacji osobników.
UNA	Patrz: unikatowość nazw.

wystąpienie kontekstu	Opis świata danego kontekstu.
zawieranie terminologii	<p>O_{x_i} zawiera się (\ll) O_{x_j}, jeżeli istnieje ścieżka z węzła odpowiadającego terminologii O_{x_i} do węzła odpowiadającego terminologii O_{x_j}:</p> $O_{x_i} \ll O_{x_j} \Leftrightarrow \exists \text{ ścieżka } p_{ij} \text{ z węzła } v_i \text{ do } v_j (f(v_i) = O_{x_i} \wedge f(v_j) = O_{x_j}).$
źródło wiedzy	<p>Źródłem wiedzy $\mathcal{KS} = \langle O, Q_O \rangle$ nazywamy system opisujący wiedzę w terminach ontologii O i dostarczający metod wnioskowania umożliwiających poprawne i kompletne odpowiadanie na zapytania sformułowane w języku zapytań Q_O.</p>

Spis definicji

DEFINICJA 2.1 FUNKCJA INTERPRETACJI W JĘZYKU \mathcal{ALC}	7
DEFINICJA 2.2 AKSJOMATY TERMINOLOGICZNE	7
DEFINICJA 2.3 ASERCJE OPISUJĄCE CECHY OSOBNIKÓW	8
DEFINICJA 2.4 FUNKCJA INTERPRETACJI W JĘZYKU \mathcal{SHION}	12
DEFINICJA 2.5 DZIEDZINA KONKRETNA \mathcal{D}	12
DEFINICJA 2.6 SYSTEM INTEGRACJI ONTOLOGII OIS	20
DEFINICJA 2.7 ONTOLOGIA LOKALNA S_l	20
DEFINICJA 2.8 ONTOLOGIA GLOBALNA \mathcal{G}	20
DEFINICJA 2.9 MODEL GLOBALNY \mathcal{O} WZGLĘDEM \mathcal{D}	21
DEFINICJA 2.10 SEMANTYKA SYSTEMU OIS	21
DEFINICJA 2.11 SPEŁNIALNOŚĆ ODWZOROWANIA $M_{\mathcal{G},S}$ PRZEZ I WZGLĘDEM \mathcal{D} W PODEJŚCIU GAV	21
DEFINICJA 2.12 SPEŁNIALNOŚĆ ODWZOROWANIA $M_{\mathcal{G},S}$ PRZEZ I WZGLĘDEM \mathcal{D} W PODEJŚCIU LAV	22
DEFINICJA 2.13 SPEŁNIALNOŚĆ ODWZOROWANIA $M_{\mathcal{G},S}$ PRZEZ I WZGLĘDEM \mathcal{D} W PODEJŚCIU NIEOGRANICZONYM	22
DEFINICJA 2.14 ZNACZENIE ZAPYTANIA W SYSTEMIE OIS	22
DEFINICJA 3.1 ONTOLOGIA \mathcal{O}	25
DEFINICJA 3.2 JĘZYK ZAPYTAŃ $Q_{\mathcal{O}}$	27
DEFINICJA 3.3 ŹRÓDŁO WIEDZY	27
DEFINICJA 3.4 ZBIÓR ODWZOROWAŃ AKSJOMATYCZNYCH \mathcal{Y}	28
DEFINICJA 3.5 ZNACZENIE TRÓJKI $(\mathcal{O}_{x_i}, \mathcal{O}_{x_j}, \mathcal{Y})$	28
DEFINICJA 3.6 HIERARCHIA TERMINOLOGII $\mathcal{H}_{\mathcal{T}}$	29
DEFINICJA 3.7 POPRAWNOŚĆ HIERARCHII TERMINOLOGII $\mathcal{H}_{\mathcal{T}}$	30
DEFINICJA 3.8 RELACJA ZAWIERANIA DLA DWÓCH TERMINOLOGII \mathcal{O}_{x_i} I \mathcal{O}_{x_j} Z $\mathcal{H}_{\mathcal{T}}$	32
DEFINICJA 3.9 SYSTEM INTEGRACJI WIEDZY KISS (<i>KNOWLEDGE INTEGRATION SUPPORT SYSTEM</i>)	32
DEFINICJA 3.10 SEMANTYKA SYSTEMU KISS	33
DEFINICJA 3.11 SEMANTYKA ZAPYTANIA $q_{\mathcal{O}_I} (\mathcal{O}_x, P_{KS}) \in Q_{\mathcal{O}_I}$ W SYSTEMIE KISS	34
DEFINICJA 3.12	35

POPRAWNOŚĆ (ANG. <i>SOUNDNESS</i>) ODPOWIEDZI $r_{O_I} (O_x, P_{KS})^{KISS}$ NA ZAPYTANIE $q_{O_I} (O_x, P_{KS})$	35
DEFINICJA 3.13 KOMPLETNOŚĆ (ANG. <i>COMPLETNESS</i>) ODPOWIEDZI $r_{O_I} (O_x, P_{KS})^{KISS}$ NA ZAPYTANIE $q_{O_I} (O_x, P_{KS})$	35
DEFINICJA 4.1 SYGNATURA DŁUGOŚCI N	39
DEFINICJA 4.2 DZIAŁANIA NA SYGNATURACH.....	40
DEFINICJA 4.3 PORZĄDEK W ZBIORZE SYGNATUR.....	40
DEFINICJA 4.4 DIALEKT KARTOGRAFICZNY $C\mathcal{L}C$	40
DEFINICJA 4.5 MAPA KONCEPTÓW DLA DIALEKTU $C\mathcal{L}C(T)$	40
DEFINICJA 4.6 ROZMIESZCZENIE OSOBNIKÓW BAZY WIEDZY K	41
DEFINICJA 5.1 ODWZOROWANIE KONCEPTU.....	45
DEFINICJA 5.2 ODWZOROWANIE ROLI	45
DEFINICJA 5.3 ODWZOROWANIE ATRYBUTU	46
DEFINICJA 5.4 ZBIÓR ODWZOROWAŃ S	46
DEFINICJA 5.5 ONTOLOGIA $O[O_x, S]$	47
DEFINICJA 5.6 POPRAWNOŚĆ ZBIORU ODWZOROWAŃ S	48
DEFINICJA 5.7 KOMPLETNOŚĆ ZBIORU ODWZOROWAŃ S	48
DEFINICJA 5.8 ODWZOROWANIE SYGNATURY	51
DEFINICJA 5.9 ZBIÓR ODWZOROWAŃ SYGNATUR.....	52
DEFINICJA 5.10 SYGNATURA MAKSYMALNEGO POKRYCIA	52
DEFINICJA 6.1 PAJĘCZYNA $P(O_{x_n}, O_{x_p})$	59
DEFINICJA 6.2 SYGNATURA NADRZĘDNA $S_N[O, P(O_{x_n}, O_{x_p}), P_{KS}]$ OSOBNIKA O DLA PAJĘCZYNY $P(O_{x_n}, O_{x_p})$ DLA ZBIORU ŹRÓDEŁ WIEDZY P_{KS}	65
DEFINICJA 6.3 SYGNATURA PODRZĘDNA $S_P[O, P(O_{x_n}, O_{x_p}), P_{KS}]$ OSOBNIKA O DLA PAJĘCZYNY $P(O_{x_n}, O_{x_p})$ DLA ZBIORU ŹRÓDEŁ WIEDZY P_{KS}	65
DEFINICJA 6.4 SYGNATURA PROSTA $S_S(O, V[\mathcal{H}_T, P_{KS}])$ OSOBNIKA O W WĘŻLE V W HIERARCHII TERMINOLOGII \mathcal{H}_T DLA ŹRÓDEŁ WIEDZY ZE ZBIORU P_{KS}	66
DEFINICJA 6.5 SYGNATURA ZŁOŻONA $S_C(O, V[\mathcal{H}_T, P_{KS}])$ OSOBNIKA O W WĘŻLE V W HIERARCHII TERMINOLOGII \mathcal{H}_T DLA ŹRÓDEŁ WIEDZY ZE ZBIORU P_{KS}	66
DEFINICJA 9.1 DEFINICJA FORMUŁY LOGICZNEJ $ALLKNOWN(P, X, Y)$	95
DEFINICJA 9.2 ODWZOROWANIE KONCEPTU $M_{C_x, I, C} = \langle C_x, I, C, q_{C_x, I, C} \rangle$	99
DEFINICJA 9.3 ODWZOROWANIE ROLI $M_{C_x, I, R} = \langle C_x, I, R, q_{C_x, I, R} \rangle$	99
DEFINICJA 9.4 ODWZOROWANIE ATRYBUTU $M_{C_x, I, A} = \langle C_x, I, A, q_{C_x, I, A} \rangle$	100
DEFINICJA 9.5 ZBIÓR ODWZOROWAŃ S	100

DEFINICJA 9.6 ONTOLOGIA $O[O_X, S]$	101
--------------------------------------------------	------------

Bibliografia

- [ANTO2004] G. Anftoniou, F. van Harmelen: *Web Ontology Language: OWL*. W: Handbook on Ontologies. Red. S. Staab, R. Studer, Springer-Verlag 2004.
- [BAAD1999] F. Baader, P. Hanschke: *A schema for integrating concrete domains into concept languages*. W Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91), str. 452–457, 1991.
- [BAAD2003] F. A. Baader, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider: *The Description Logic Handbook: Theory, implementation, and applications*, Cambridge University Press, 2003.
- [BECH2003] S. Bechhofer: *The DIG Description Logic Interface: DIG/1.1*, University of Manchester, 2003.
- [BERN2001] T. Berners-Lee, J. Hendler, O. Lassila: *The Semantic Web*, maj 2001; Scientific American Magazine.
- [BOUQ2005] P. Bouquet, Ch. Ghidini, F. Giunchiglia, E. Blanzieri: *Theories and uses of context in knowledge representation and reasoning*. *Journal of Pragmatics*, 35(3), str. 455-484, 2003. Intelligence, Third International Atlantic Web Intelligence Conference, Springer 2005. str. 163-169.
- [BRAC2004] Brachman R. J., Levesque H.: *Knowledge Representation and Reasoning*, Morgan Kaufmann Publishers, 2004.
- [BUVA1995] Buvač S., Mason I. A.: *Mathematics of Contexts*. *Fundamentae Informaticae*, 23(3), 1995.
- [CALV2001] D. Calvanese, D. Giacomo, M. Lenzerini: *Ontology of integration and integration of ontologies*, Proceedings of the International Workshop on Description Logics, 2001.
- [CALV2002] D. Calvanese, D. Giacomo, M. Lenzerini: *A framework for ontology integration*, (2002), Proc. of the 1st Semantic Web Working Symposium at the Emerging Semantic Web, str. 201-214.
- [CHAWA1994] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. *The TSIMMIS Project: Integration of heterogeneous information sources*. W: Proceedings of the 10th Meeting of the Information Processing Society of Japan, str. 7–18, Tokyo, Japan, Październik 1994.
- [DAVE1998] T. Davenport, L. Prusak: *Working knowledge*, Harvard Business Press, Boston, 1998.
- [DIG2.0] *DIG 2.0: The DIG Description Logic Interface*, DIG Working Group Note September 2006, <http://dig.cs.manchester.ac.uk/>.
- [DIG2.0QI] *DIG 2.0 Proposal for a Query Interface*, <http://www.sts.tu-harburg.de/~al.kaplunova/dig-query-interface.html>.

- [DIGUT] *DIGUT Interface Version 1.3*. Raport grupy KMG@GUT, 2005, dostępny pod adresem:
http://km.pg.gda.pl/km/digut/1.3/DIGUT_Interface_1.3.pdf.
- [DOAN2004] A. Doan, J. Madhavan, P. Domingos, A. Halevy: *Ontology Matching: A Machine Learning Approach*. W: Handbook on Ontologies. Red. S. Staab, R. Studer, Springer-Verlag 2004.
- [DONI1992] Donini F. M., Lenzerini M., Nardi D., Nutt W., Schaerf A.: *Adding epistemic operators to concept languages*. W: Proc. of the 3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'92), s. 342–353. Morgan Kaufmann, Los Altos, 1992.
- [DONI1998] Donini F. M., Lenzerini M., Nardi D., Nutt W., Schaerf A.: *An epistemic operator for description logics*. Artificial Intelligence, 100(1–2):225–274, 1998.
- [EHRIG2004] M. Ehrig, P. Haase, N. Stojanovic: *Similarity for ontologies - a comprehensive framework*. W: Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability, at PAKM 2004, DEC 2004.
- [GADO1999] A. M. Gadomski: *Meta-Ontological Assumptions: Information, Preferences and Knowledge universal interrelations (IPK cognitive architecture)*,
<http://erg4146.casaccia.enea.it/wwwerg26701/gad-dict.htm>,
 strona od 1999, data ostatniej modyfikacji grudzień 2005, MKEM Server.
- [GHID2001] Ghidini Ch., Giunchiglia F.: *Local Model Semantics, or Contextual Reasoning = Locality + Compatibility*. Artificial Intelligence, 127(2), str. 221-259, 2001.
- [GIUN1993] Giunchiglia F.: *Contextual Reasoning. Epistemologia, special issue on I Linguaggi e le Macchine*, XVI, str. 345-364, 1993.
- [GIUN1994] Giunchiglia F., Serafini L.: *Multilanguage hierarchical logics or: how we can do without modal logics*. Artificial Intelligence, 65(1), str. 29-70, 1994.
- [GOCZ2004] Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: *Problematyka zarządzania wiedzą w systemach typu e-health*. W: Inżynieria oprogramowania. Nowe wyzwania. Red. Górski J., Wardziński A. Warszawa WNT 2004. Gdańsk 2004. (str. 357-371).
- [GOCZ2005] K. Goczyła, W. Waloszek, M. Zawadzki, T. Grabowska: *The Cartographer Algorithm for Processing and Querying Description Logics Ontologies*. W: LNAI 3528: Advances In Web Intelligence, AWIC 2005, str. 163–169, 2005.
- [GOCZ2005a] K. Goczyła, T. Zawadzka: *Przetwarzanie zapytań w rozproszonej bazie wiedzy opartej na logice opisowej*. W: Bazy Danych - Modele, Technologie, Narzędzia, Ed. S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek, WKŁ 2005, Roz. XXIV, str. 199-206.
- [GOCZ2005b] K. Goczyła, T. Grabowska, W. Waloszek, M. Zawadzki: *Inference Mechanisms for Knowledge Management System in E-health Environment*, IOS Press 2005, str. 418 – 423.

- [GOCZ2005c] Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: *Hybryd Architecture of DL Knowledge Base in KaSeA*. International Workshop on Description Logics, 2005, str. 216.
- [GOCZ2005d] Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: *Cartographic Approach to Knowledge Representation and Management in KaSeA*. International Workshop on Description Logics, 2005, str. 217.
- [GOCZ2005e] Goczyła K., Waloszek W., Zawadzka T., Zawadzki M.: *Designing World Closures for Knowledge-Based System Engineering*, IOS Press 2005, str. 271 – 282.
- [GOCZ2005f] Goczyła K., Grabowska T.: *Metoda ELPAR łączenia ontologii oparta na ich kartograficznej reprezentacji*. W: Technologie Przetwarzania Danych, Red. T. Morzy, H. Biński, Wydawnictwo Politechniki Poznańskiej 2005, str. 26-38.
- [GOCZ2006] K. Goczyła, T. Grabowska, W. Waloszek, M. Zawadzki: *The Knowledge Cartography — A New Approach to Reasoning over Description Logics Ontologies*. W: LNCS 3831: SOFSEM 2006: Theory and Practice of Computer Science, str. 293–302, Springer, 2006.
- [GOCZ2006a] K. Goczyła, T. Zawadzka,: *Ontologie w Sieci Semantycznej*, Studia Informatica, Vol. 27 (2006) nr 2 (67). Wydawnictwo Politechniki Śląskiej, Gliwice, str. 65-92.
- [GOCZ2006b] Goczyła, K., Zawadzka, T., Zawadzki, M., *Managing Data from Heterogeneous data Sources using Knowledge Layer*, Software Engineering Techiques: Design for Quality, IFIP International Federation for Information processing, Vol. 227, K. Sacha (red.), Boston, Springer, 2006, str. 300-312.
- [GOCZ2006c] K. Goczyła, T. Zawadzka: *Zależności między ontologiami i ich wpływ na problem integracji ontologii*. W: Bazy Danych, Struktury, Algorytmy, Metody - architektura, metody formalne i eksploracja danych : praca zbiorowa pod red. S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek - Warszawa : WKŁ, 2006. - s. 331-340.
- [GOCZ2006d] Goczyła K., Waloszek W.: *Tworzenie map konceptów dla ontologii opartych na logice opisowej*, w: Bazy Danych Struktury, Algorytmy, Metody, red. Kozielski S., Małysiak B., Kasprowski P., Mrozek D., s. 319–329, WKŁ, 2006.
- [GOCZ2006e] Goczyła K., Waloszek A., Waloszek W., Zawadzka T., Zawadzki M.: *Ontological queries supporting decision process in KaSea system*. W: EJC 2006 : Proceedings of the 16th European - Japanese Conference on Information Modelling and Knowledge Bases Trojanovice, Czech Republic 29 May - 2 June, 2006 / eds: Y. Kiyoki, H. Kangassalo, M. Duzi. - Ostrava : Dept. of Computer Sci., Tech. Univ. Ostrava, 2006. - s. 16-28.
- [GOCZ2007] Goczyła K. Zawadzka T., Zawadzki M.: *Wnioskowanie z danych zapisanych w zewnętrznych źródłach w systemie zarządzania wiedzą*. W: Bazy Danych, Nowe technologie – Architektura, metody formalne i zaawansowana analiza danych: praca zbiorowa pod red. S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek. Warszawa: WKŁ 2007. s. 283-294.

- [GOCZ2007a] Goczyła K., Waloszek A., Waloszek W.: *Hierarchiczny podział przestrzeni ontologii na konteksty*, w: Bazy danych. Nowe technologie, red. Kozielski S., Małysiak B., Kasprowski P., Mrozka D., s. 247-260, WKŁ, 2007.
- [GOCZ2007b] Goczyła K., Waloszek A., Waloszek W.: *Contextualization of a DL Knowledge Base*, w: Proceedings of the International Workshop on Description Logics, s. 291-298, 2007.
- [GOCZ2007c] Goczyła K., Waloszek A., Waloszek W.: *Wnioskowanie z dziedzin konkretnych w podejściu kartograficznym*. W: Bazy danych. Nowe technologie, red. Kozielski S., Małysiak B., Kasprowski P., Mrozka D., s. 261-272, WKŁ, 2007.
- [GOCZ2008] Goczyła K., Zawadzka T.: *Metody wnioskowania kontekstowego z danych zapisanych w zewnętrznych źródłach*, w: Bazy danych. Rozwój metod i technologii, red. Kozielski S., Małysiak B., Kasprowski P., Mrozek D., s. 269-281, WKŁ, 2008.
- [GOCZ2008a] Goczyła K., Zawadzka T.: *Algorithms for QueryProcessing In a Distributed Knowledge Integration System*. W: Proceedings of the 2008 1st International Conference on Information Technology, IT 2008, red. Stepnowski A., Moszyński M., Kochanski T., Dąbrowski J., s. 27-30.
- [GOCZ2008b] Goczyła K., Zawadzka T., Zawadzki M.: *Model systemu zarządzania wiedzą z uwzględnieniem aspektów wiarygodności*. W: Bazy Danych, Rozwój metod i technologii, red S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek, Warszawa WKŁ, 2008, str. 283-293.
- [GOCZ2008c] Goczyła K., Zawadzka T., Zawadzki M.: *Defining Trustworthiness in Semantic Web by Ontological Assertions*. W: Proceedings of the 2008 1th International Conference on Information Technology, IT 2008, red. Stepnowski A., Moszyński M., Kochanski T., Dąbrowski J., s. 23-26.
- [GRAB2004] T. Grabowska, M. Zawadzki: *Zastosowanie relacyjnych baz danych i logiki opisowej w systemach integracji danych*. W: Technologie Informacyjne, Gdańsk 2004, str. 907 – 914.
- [GRAB2004a] T. Grabowska: *Wykorzystanie ontologii wzbogaconych o reguły logiczne w systemach integracji danych*. W: Multimedialne i sieciowe systemy informacyjne. Red. Cz. Daniłowicz. Wrocław 2004, str. 37-46.
- [GRUB1993] T. Gruber: *A translation approach to portable ontology specifications*, Knowledge Acquisition, 5(2): 199-220, 1993.
- [HAAR2001] Haarslev V., Möller R.: *RACER System Description*, w: Proceedings of IJCAR 2001, LNAI 2083, s. 701-706, Springer-Verlag, 2001 (patrz też: <http://www.racer-systems.com/>).
- [HON2008] *Health on the Net Foundation*, <http://www.hon.ch/>.
- [HORR1999] Horrocks I., Sattler U., Tobies S.: *Practical reasoning for expressive description logics*. W: Proceedings of LPAR'99, LNAI 1705, s.161-180, Springer-Verlag, 1999.
- [HTML1999] *HTML 4.01 Specification*, W3C Recommendation 24 December 1999, <http://www.w3.org/TR/html4/>.

- [JENN2001] N.R. Jennings: *An agent-based approach for building complex software systems*. W: Communications of the ACM, 44(4), 2001, 35-41.
- [KMG] Knowledge Management Group at Gdańsk University of Technology, <http://km.pg.gda.pl/kmg/>.
- [LATT1998] V. Lattes, M. Rousset: *The use of CARIN language and algorithms for Information Integration: the PICSEL project*, W: Proceedings of the ECAI-98 Workshop on Intelligent Information Integration, 1998.
- [LEVY1998] A. Y. Levy: *The Information Manifold Approach to Data Integration*, IEEE Intelligent Systems, numer 13, 1998.
- [LEVY2000] A. Y. Levy: *Logic based techniques in data integration*, Kluwer Publishers 2000.
- [LEVY2001] A. Y. Levy: *Answering queries using views: a survey*. In: VLDB Journal 10, S. 270-294, 2001.
- [MCBR2004] B. McBride: *The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS*. W: In: Handbook on Ontologies. Red. S. Staab, R. Studer, Springer-Verlag 2004.
- [OWL2004] *OWL Web Ontology Language Guide*, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/owl-guide/>.
- [OWL1.12006] *OWL 1.1 Web Ontology Language*, http://owl1_1.cs.manchester.ac.uk/overview.html.
- [OWLQLSO] *OWL-QL Syntax overview*: <http://ksl.stanford.edu/projects/owl-ql/syntax.shtml>.
- [OWLQL2003] Fikes, R., Hayes, P., Horrocks, I.: *OWL-QL - A Language for Deductive Query Answering on the Semantic Web*. Knowledge Systems Laboratory, Stanford University, Stanford, CA, 2003, http://ksl.stanford.edu/KSL_Abstracts/KSL-03-14.html.
- [OWLS2004] *OWL-S: Semantic Markup for Web Services*, W3C Member Submission 22 November 2004, <http://www.w3.org/Submission/OWL-S/>.
- [PINT1999] S. Pinto, A. Gomez-Perez, J. Martins: *Some Issues on Ontology Integration*, Proceedings of the JCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, 1999.
- [PINT2001] H. S. Pinto, J. P. Martins: *A Methodology for Ontology Integration*. W: Proceedings of the First International Conference on Knowledge Capture, ACM Press.
- [PIPS2007] *Personalised Information Platform for Life and Health Services*, http://smart.eservices4life.org/ideas/demo_def/index.html.
- [PIPS2008] *Materiały informacyjne seminarium PIPS*, Politechnika Gdańska, Katedra Inżynierii Oprogramowania, 2008, <http://www.eti.pg.gda.pl/katedry/kio/?akcja=wiecej&wiadomosc=4919>.
- [POTT2001] R. Pottinger, A.Y. Levy: *MiniCon: A Scalable algorithm for answering queries using views*. In: VLDB Journal, 2001.
- [PREE2001] A. Preece, D. H. Sleeman, A. N. Flett, D. Curry, N. Meaney, P. Perry: *Better Knowledge Management through Knowledge Engineering*. W: IEEE Intelligent Systems, 14(1), 26-36.

- [RACER] *RacerPro User's Guide*: <http://www.racer-systems.com/products/racerpro/manual.phtml>.
- [RDFaP] *RDFa Primer, Bridging the Human and Data Webs*, <http://www.w3.org/TR/xhtml-rdfa-primer/>.
- [RDFaSP] *RDFa in XHTML: Syntax and Processing, A collection of attributes and processing rules for extending XHTML to support RDF*, <http://www.w3.org/TR/rdfa-syntax/>.
- [RDF2004] *RDF/XML Syntax Specification (Revised)*, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [RDFS2004] *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/rdf-schema/>.
- [REIT78] R. Reiter: *On reasoning by default*, w: Proceedings of TINLAP-2, Theoretical Issues in Natural Language Processing-2, s. 210–218, 1978.
- [RUSS2003] S. J. Russel, P. Norvig: *Artificial Intelligence. A Modern Approach*. Second Edition. Pearson Education International. 2003.
- [SIRI2004] Sirin E., Parsia B.: *Pellet: An OWL DL reasoner*, w: Proceedings of the International Workshop on Description Logics, s. 212–213, 2004 (patrz też: <http://pellet.owldl.com/>).
- [SWEB2001] *Semantic Web Activity*, <http://www.w3.org/2001/sw/>.
- [STAA2004] S. Staab, R. Studer: *Handbook on Ontologies*. International Handbooks on Information Systems, Springer, 2004.
- [SUNA2002] N. Sunasee, D. Sewry: *A theoretical Framework for Knowledge Management* Rhodes University, Proceedings of SAICSIT, 2002.
- [SWRL2004] *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, W3C Member Submission 21 May 2004, <http://www.w3.org/Submission/SWRL/>.
- [TOKA1970] J. Tokarski: *Słownik wyrazów obcych*, Państwowe Wydawnictwo Naukowe, Warszawa, 1970.
- [ULLMAN1997] Jeffrey Ullmann: *Information Integration Using Logical Views*. In: Proceedings of the 6th International Conference on Database Theory. Springer, 1997. str. 19-40.
- [URIRFC2396] *Uniform Resource Identifiers (URI): Generic Syntax*, <http://tools.ietf.org/html/rfc2396>.
- [URLRFC1738] *Uniform Resource Locators (URL)*, <http://tools.ietf.org/html/rfc1738>.
- [WALO2005] Waloszek W.: *Cartographic Method of Knowledge Representation in KaSeA*, Technologie Przetwarzania Danych, Wydawnictwo Politechniki Poznańskiej, 2005, str. 14-25.
- [WALO2008] Waloszek W.: *Rozprawa doktorska: Metody strukturalnej analizy ontologii opartych na logice opisowej*, Gdańsk 2008.
- [WITT2000] Wittem I. H., Frank E.: *Data Mining. Practical Machine Learning Tools and Techniques, with Java Implementations*. Morgan Kaufmann Publisher 2000.
- [XML2006] *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation 16 August 2006, <http://www.w3.org/TR/xml/>.

- [XMLSCHEMA2004] *XML Schema*, W3C Recommendation 28 October 2004, <http://www.w3.org/XML/Schema>.
- [ZAWA2007] Zawadzki M.: *Definiowanie wiarygodności w Sieci Semantycznej metodą ontologicznego opisu asercji*. W: Bazy Danych, Nowe Technologie, Architektura, metody formalne i zaawansowana analiza danych, red S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek, Warszawa WKŁ, 2007, str. 273-282.
- [ZAWA2007a] Zawadzka T.: *EQM Merging Query Responses_1.0.pdf*, raport roboczy projektu PIPS, 2007.